B.Tech Information Technology

1.3.2 List of Courses that include experiential learning through project work/field work/internship

2020-2021

S.No	Name of the Course and Code that include Experential Learning through (project work/field visit/internship)
1	Mini Project - MATHEMATICS-I/T101
2	Mini Project - PHYSICS/T102
3	Mini Project - CHEMISTRY /T103
4	Lab Visit - BASIC ELECTRICAL AND ELECTRONICS ENGINEERING/T104
5	Lab Visit - ENGINEERING THERMODYNAMICS/T105
6	Mini Project - COMPUTER PROGRAMMING/T106
7	Mini Project - COMPUTER PROGRAMMING LAB/P101
8	Mini Project - BASIC ELECTRICAL AND ELECTRONICS ENGINEERING LAB/P103
9	Mini Project - ENGINEERING GRAPHICS/P102
10	Mini Project - MATHEMAICS-II/T107
11	Mini Project - MATERIAL SCIENCE/T108
12	Mini Project - ENVIORNMENTAL SCIENCE/T109
13	Mini Project - BASIC CIVIL AND MECHANICAL ENGINEERING/T110
14	Mini Project - ENGINEERING MECHANICS/T111
15	Mini Project - ELECTRONICS DEVICES AND CIRCUITS/ITT32
16	Mini Project - DATA STRUCTURES/ITT33
17	Mini Project - OBJECT ORIENTED PROGRAMMING/ITT34
18	Mini Project - DIGITAL SYSTEM DESIGN/ITT35
19	Mini Project - DATA STRUCTURES LABORATORY/ITP31
20	Mini Project - ELECTRONICS DEVICES AND CIRCUITS LAB/ITP32
21	Mini Project - DIGITAL SYSTEM DESIGN LABORATORY/ITP33
22	Mini Project - COMMUNICATION ENGINEERING-I/ITT42
23	Mini Project - DESIGN AND ANALYSIS OF ALGORITHMSITT43
24	Mini Project - MICROPROCESSORS AND MICROCONTROLLERS/ITT44
25	Mini Project - JAVA PROGRAMMING/ITT45
26	Mini Project - SYSTEM SOFTWARE/ITT46
27	Mini Project - ALGORITHM LABORATORY/ITP41
28	Mini Project - MICROPROCESSOR AND MICROCONTROLLERS LAB/ITP42
29	Mini Project - JAVA LABORATORY/ITP43

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY Puducherry 605 107

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

30	Mini Project - COMMUNICATION ENGINEERING-II/ITT51
31	Mini Project - SOFTWARE ENGINEERING/ITT52
32	Mini Project - OPERATING SYSTEMS/ITT53
33	Mini Project - DATABASE MANAGEMENT SYSTEMS/ITT54
34	Mini Project - THEORY OF COMPUTATION/ITT55
35	Mini Project - COMPUTER HARDWARE AND TROUBLE SHOOTING/ITE51
36	Mini Project - COMMUNICATION ENGINEERING LABITP51
37	Mini Project - OPERATING SYSTEMS LABORATORY/ITP52
38	Mini Project - DATABASE MANAGEMENT SYSTEMS LAB/ITP53
39	Mini Project - COMPUTER NETWORK/ITT61
40	Mini Project - WEB TECHNOLOGY/ITT62
41	Mini Project - ARTIFICIAL INTELLIGENCE/ITT63
42	Mini Project - INFORMATION CODING TECHNIQUE/ITT64
43	Mini Project - OBJECT ORIENTED ANALYSIS AND DESIGN/ITE66
44	Mini Project - USER INTERFACE DESIGN/ITE68
45	Mini Project - COMPUTER NETWORK LABORATORY/ITP61
46	Mini Project - WEB TECHNOLOGY LABORATORY/ITP62
47	MINI PROJECT LAB/ITP63
48	Mini Project - MOBILE COMPUTING/ITT71
49	Mini Project - WEB SERVICE AND XML/ITT72
50	Mini Project - CRYPTOGRAPHY AND NETWORK SECURITY/ITT73
51	Mini Project - SOFTWARE TESTING/ITE72
52	Mini Project - BIG DATABASE/ITE79
53	Mini Project - MOBILE COMPUTING LABORATORY/ITP71
54	Mini Project - WSX LABORATORY/ITP72
55	PROJECT WORK (PHASE I)/ITP73
56	Mini Project - PROFESSIONAL ETHICS/ITT81
57	Mini Project - DATA MINING/ITE83
58	PROJECT WORK PHASE II/ITP81

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY Puducherry 605 107

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 10

T 104 - BASIC ELECTRICAL AND ELECTRONICS ENGINEERING

COURSE OBJECTIVE

- *To understand and gain basic knowledge about magnetic and electrical circuits, single phase and three phase power measurement and the operating principles of stationary and rotating machines*
- *To understand the basic operation, functions and applications of PN junction diode, transistor, logic gates and flip flops.*
- *To gain knowledge on various communication systems and network models and the use of ISDN*

COURSE OUTCOME

On successful completion of the module students will be able to:

- *Will gain basic knowledge about magnetic and electrical circuits, single phase and three phase power measurement and the operating principles of stationary and rotating machines*
- *Design and conduct experiment, as well as to analyze the basic operation, functions and applications of PN junction diode, transistor, logicgates and flip flops.*
- *Identify and analyze various communication systems and network models and the applications of ISDN*

PART A – Electrical

UNIT – I

DC Circuit: Definition of Voltage, Current, Power & Energy, circuit parameters, Ohm's law, Kirchoff's law & its applications – Simple Problems - Division of current in Series& parallel circuits - star/delta conversion - Node and mesh methods of analysis of DC circuits. **(10)**

UNIT – II

AC Circuit: Concepts of AC circuits – rms value, average value, form and peak factors – Simple RLC series circuits – Concept of real and reactive power – Power factor - Introduction to three phase system - Power measurement by two wattmeter method. **(10)**

UNIT – III

Electrical Machines and Power Plants: Law of Electromagnetic induction, Fleming's Right & Left hand rule - Principle of DC rotating machine, Single phase transformer and single phase induction motor(Qualitative approach only) - Simple layout of thermal and hydro generation (block diagram approach only).Fundamentals of fuses and circuit breakers. **(10)**

PART – B – Electronics

UNIT – IV

Electronic Circuit: V-I Characteristics of diode - Half-wave rectifier and Full-wave rectifier – with and without capacitor filter - Transistor - Construction & working - Input and output characteristics of CB and CE configuration - Transistor as an Amplifier - Principle and working of Hartley oscillator and RC phase shift oscillator - Construction and working of JFET & MOSFET. **(10)**

UNIT – V

Digital Electronics: Boolean algebra – Reduction of Boolean expressions - De-Morgan's theorem – Logic gates -Implementation of Boolean expressions - Flip flops - RS, JK, T and D. Combinational logic - Half adder, Full adder and Subtractors. Sequential logic - Ripple counters and shift registers. **(10)**

UNIT – IV

Communication and Computer Systems: Model of communication system – Analog and digital – Wired and wireless channel. Block diagram of various communication systems – Microwave, satellite, optical fiber and cellular mobile system. Network model – PAN, LAN, MAN and WAN – Circuit and packet switching – Overview of ISDN. **(10)**

Text Books:

1. Kothari D P and Nagrath I J , Basic Electrical Engineering , Tata McGraw Hill,2009. (For Units I to III)
2. Rajendra Prasad , " Fundamentals of Electronic Engineering", Cengage learning, New Delhi, First Edition, 2011 (For Unit IV)
3. Morris Mano, "Digital design", PHI Learning, Fourth Edition, 2008 (For Unit V)
4. Wayne Tomasi, "Electronic Communication Systems- Fundamentals Theory Advanced", Sixth Edition, Pearson Education, 2004. (For Unit VI)

Reference Books:

1. R.Muthusubramaniam, S.Salivahanan and K.A. Mureleedharan, Basic Electrical Electronics and Computer Engineering, Tata McGraw Hill, 2004..
2. J.B.Gupta, A Course in Electrical Power, Katson Publishing House, New Delhi, 1993.
3. David. A. Bell, "Electronic Devices and Circuits", PHI Learning Private Ltd, India, Fourth Edition, 2008

4. Donald P Leach, Albert Paul Malvino and Goutam Saha, "Digital Principles and Applications," 6th edition, Tata McGraw Hill Publishing Company Ltd., New Delhi,2008.
5. S.K. Sahdev, Fundamentals of Electrical Engineering and Electronics, Dhanpat Rai & Co, 2013.
6. Jacob Millman and Christos C. Halkias, "Electronic Devices and Circuits" Tata McGraw Hill,2008
7. R.L. Boylestad and L. Nashelsky, "Electronic Devices and Circuit Theory", PHI Learning Private Limited, Ninth Edition, 2008.
8. M.S.Sukhija and T.K.Nagsarkar, " Basic Electrical and Electronics Engineering", Oxford University Press, 2012

KIRCHHOFF'S VOLTAGE LAW & KIRCHHOFF'S CURRENT LAW USING FIRST ORDER DIFFERENTIAL EQUATION

PLACE OF VISIT : BASIC ELECTRICAL & ELECTRONICS LAB AT MVIT

SUBJECT: BASIC ELECTRICAL & ELECTRONICS ENGINEERING

I YEAR /I SEM IT – C SEC STUDENTS

1. INTRODUCTION

An industrial visit to Basic Electrical was arranged on 10th March 2021. First yearInformation Technology students were taken to the visit under the guidance of Dr.K.Sedhuraman Asst. Professor, Dept of Electrical Engineering. He explained well about the circuit containing an inductance L or a capacitor C and resistor R with current and voltage and said the voltage variable given by differential equation. The general solution of differential equation represent the complete response of network . A circuit containing an inductance L or a capacitor C and resistor R with current and voltage variable given by differential equation of the same form.. Thus students gained information about voltage-current relation across inductor as it is given by the differential equation which leads to Kirchhoff's laws to contain derivatives. Hence to modeling of the variation of a physical quantity such as current ,voltage would result in differential equations. The visit becomes useful for students development and has increased the span of students' knowledge.

2. OBJECTIVE OF THE VISIT

The Objective of visit was, students are able

- To understand the components relating to voltage variables.
- To frame Khiroff,s voltage and current law using differential equations.

3. PURPOSE OF VISIT

Main purpose for this visit was to give the practical knowledge about the framing Khiroff,s voltage and current law using differential equations

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,

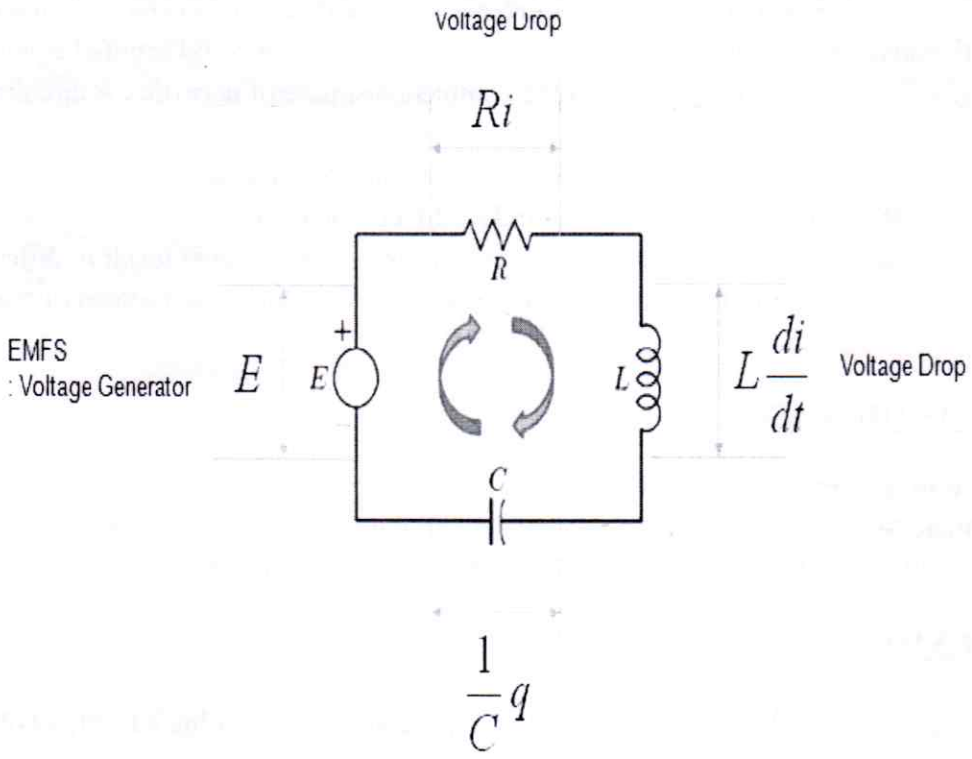
In general a circuit containing an inductance L or a capacitor C and resistor R with current and voltage variable given by differential equation. The general solution of differential equation represent the complete response of network . The v-i relation for an inductor or capacitor is a differential. A circuit containing an inductance L or a capacitor C and resistor R with current and voltage variable given by differential equation of the same form. It is a linear first order differential equation with constant coefficient when the value of R,L,C are constant. L and C are storage elements. Thus voltage-current relation across inductor is given by the differential equation which leads to Kirchhoff's laws to contain derivatives. Hence to modeling of the variation of a physical quantity such as current ,voltage would result in differential equations

Kirchhoff's voltage law

Kirchhoff's voltage law states that the sum of the voltage drops across an inductor, L dI/dt, and across a resistor, IR, must be the same as the voltage source, E(t), applied to the circuit. The resulting ODE model is LdI/dt+IR=E(t).

COMPONENTS

RESISTOR ,INDUCTOR ,VOLATAGE GENERATOR ,BATTERY



The inductor equation tells us

$$v = \frac{di}{dt}$$

$$(+E) + (-Ri) + (-L\frac{di}{dt}) + (-\frac{1}{C}q) = 0$$

$$E = Ri + L\frac{di}{dt} + \frac{1}{C}q = 0$$

Taking diff on both sides

$$\frac{dE}{dt} = R\frac{di}{dt} + L\frac{d}{dt}\frac{di}{dt} + \frac{1}{C}\frac{dq}{dt}$$

$$\frac{dE}{dt} = R\frac{di}{dt} + L\frac{d^2i}{dt^2} + \frac{1}{C}i$$

Outcomes :

Through this visit, students gained information and practical knowledge about various components involved in framing Kirchhoff's voltage and current using first order differential equation .

A. Venkat 15.03.21

STAFF INCHARGE

COORDINATOR

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY ★ PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

DEPARTMENT OF GENERAL ENGINEERING
LAB VISIT TO BEEE LAB
STUDENTS NAME LIST - IT C SEC

Sl. No	Enroll.no	Name of the Students
1	20TH0151	ABDUL NAIMSA A
2	20TH0152	ABHINAYA A
3	20TH0154	ABINAYA SHRI.R
4	20TH0155	ADHAVAN.M
5	20TH0156	AKSHAYA G
6	20TH0157	ALJASSIRA.J
7	20TH0158	ASHIK A
8	20TH0159	ASHWANTHRI G
9	20TH0160	BALAJI.D
10	20TH0161	BHARATH KUMAR T
11	20TH0162	BUVANESWARAN P
12	20TH0163	CHANDRABOSE S
13	20TH0165	DEEPIKA.R
14	20TH0166	DEINESSH A
15	20TH0167	DHIVAGAR G
16	20TH0168	FADIL A B
17	20TH0169	GANESH C
18	20TH0170	GOKULRAJAN M V
19	20TH0171	GOWRI.E
20	20TH0173	HARIPRIYA Y
21	20TH0174	HARISH.S
22	20TH0177	JAYABHARATHI J
23	20TH0178	JEGATHESH S
24	20TH0181	KAMALI.V
25	20TH0182	KAVIARASAN.A
26	20TH0186	LOGESHWARI E
27	20TH0190	MOHANA PRIYA M
28	20TH0192	MUHAMMED NABIL K A
29	20TH0195	NATHAN S
30	20TH0196	NAVINKUMAR.I
31	20TH0199	PRASANNARAJ.G

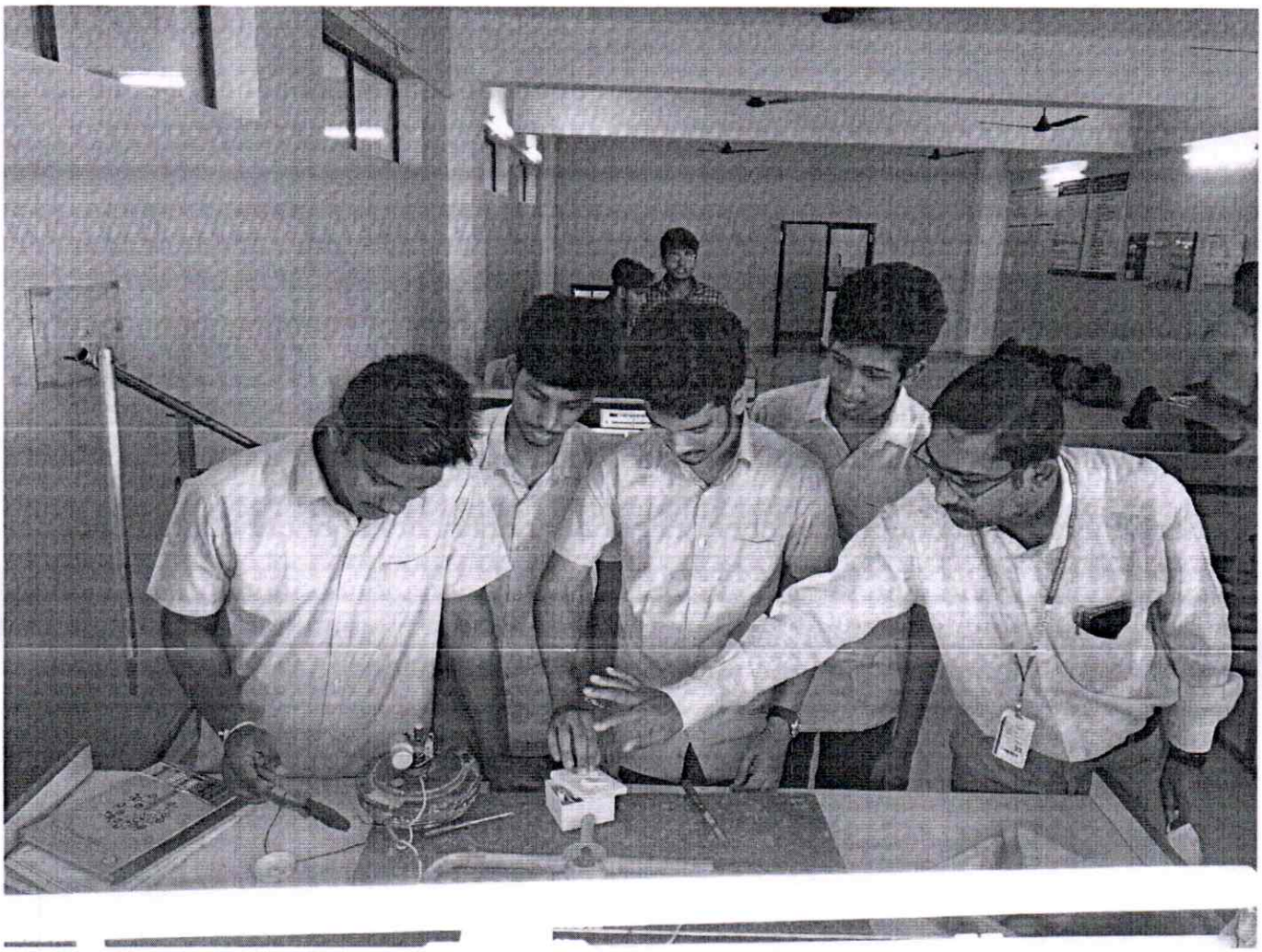
PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY

32	20TH0200	PRIYADHARSHINI R
33	20TH0202	RADHIGA R
34	20TH0203	RAJASEKAR R
35	20TH0204	RAMYA R
36	20TH0205	RENGA RITHISH G
37	20TH0206	RESHMI M
38	20TH0208	REVATHY G
39	20TH0209	RISWANA M R
40	20TH0210	SAKTHI MURUGAN P
41	20TH0211	SAKTHIBAMA E
42	20TH0213	SANJAY R
43	20TH0217	SHAGUL HAMEED S
44	20TH0218	SHAHAE MEERAN M A
45	20TH0222	SRIDEVI R
46	20TH0226	SRUTHI B
47	20TH0228	SUBATHRADEVI.P
48	20TH0229	SYED IBRAHIM M I
49	20TH0233	VASANTH S
50	20TH0236	VIGNEASWARAN.S
51	20TH0238	VISHAL.N
52	20TH0240	YUVAPRIYA.R
53	20TH0241	YUVARAJ K `
54	20TH0242	YUVARAJA R

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY ★ PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY

PHOTOS



STUDENTS MODELLING CIRCUITS AT BEEE LAB

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

Kalitheerthalkuppam, Madagadipet, Puducherry - 605 107

DEPARTMENT OF GENERAL ENGINEERING
LAB VISIT TO BEEE LAB

ATTENDANCE - IT C SEC ON 10/03/2022

Sl. No	Enroll.no	Name of the Students	Signature
1	20TH0151	ABDUL NAIMSA A	Abdul Naimsa
2	20TH0152	ABHINAYA A	Abhinaya.A
3	20TH0154	ABINAYA SHRI.R	Abinaya Shri.R
4	20TH0155	ADHAVAN.M	Adhavan
5	20TH0156	AKSHAYA G	Akshaya.G.
6	20TH0157	ALJASSIRA.J	Aljassira
7	20TH0158	ASHIK A	Ashik
8	20TH0159	ASHWANTHRI G	Ashwanthri
9	20TH0160	BALAJI.D	Balaji.
10	20TH0161	BHARATH KUMAR T	Bharath Kumar
11	20TH0162	BUVANESWARAN P	G Kishore
12	20TH0163	CHANDRABOSE S	Chandrabose.
13	20TH0165	DEEPIKA.R	Deepika.R
14	20TH0166	DEINESSH A	Dhinesh
15	20TH0167	DHIVAGAR G	Dhivagar
16	20TH0168	FADIL A B	Fadil.A.B.
17	20TH0169	GANESH C	Ganesh C.
18	20TH0170	GOKULRAJAN M V	Gokulrajan
19	20TH0171	GOWRI.E	Gowri.
20	20TH0173	HARIPRIYA Y	Haripriya Y
21	20TH0174	HARISH.S	Harish.S
22	20TH0177	JAYABHARATHI J	Jayabharathi
23	20TH0178	JEGATHESH S	Jegathesh
24	20TH0181	KAMALI.V	Kamali.V
25	20TH0182	KAVIARASAN.A	Kaviarasan
26	20TH0186	LOGESHWARI E	Logeshwari.E.
27	20TH0190	MOHANA PRIYA M	Mohanapriya.M.
28	20TH0192	MUHAMMED NABIL K A	Nabil

PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR

29	20TH0195	NATHAN S	
30	20TH0196	NAVINKUMAR.I	
31	20TH0199	PRASANNARAJ.G	
32	20TH0200	PRIYADHARSHINI R	
33	20TH0202	RADHIGA R	
34	20TH0203	RAJASEKAR R	
35	20TH0204	RAMYA R	
36	20TH0205	RENGA RITHISH G	
37	20TH0206	RESHMI M	
38	20TH0208	REVATHY G	
39	20TH0209	RISWANA M R	
40	20TH0210	SAKTHI MURUGAN P	
41	20TH0211	SAKTHIBAMA E	
42	20TH0213	SANJAY R	
43	20TH0217	SHAGUL HAMEED S	
44	20TH0218	SHAHAE MEERAN M A	
45	20TH0222	SRIDEVI R	
46	20TH0226	SRUTHI B	
47	20TH0228	SUBATHRADEVI.P	
48	20TH0229	SYED IBRAHIM M I	
49	20TH0233	VASANTH S	
50	20TH0236	VIGNEASWARAN.S	
51	20TH0238	VISHALI.N	
52	20TH0240	YUVAPRIYA.R	
53	20TH0241	YUVARAJ K	
54	20TH0242	YUVARAJA R	

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

T 105 - ENGINEERING THERMODYNAMICS

COURSE OBJECTIVE

- *To understand the basics of the thermodynamic principles and establish the relationship of these principles to thermal system behaviors*
- *To develop methodologies for predicting the system behavior and establish the importance of laws of thermodynamics applied to energy systems*
- *To explain the role of refrigeration and heat pump as energy systems and develop an intuitive understanding of underlying physical mechanism and a mastery of solving practical problems in real world*

COURSE OUTCOME

On successful completion of the module students will be able to:

- *Apply knowledge of mathematics, science and engineering to understand the basics of the thermodynamic principles and establish the relationship of these principles to thermal system behaviors*
- *Design and conduct experiment, as well as to analyze and develop methodologies for predicting the system behavior and understand the importance of laws of thermodynamics applied to energy systems*
- *Identify and analyze role of refrigeration and heat pump as energy systems and develop an intuitive understanding of underlying physical mechanism and a mastery of solving practical problems in real world*

UNIT – I

Basic Concepts and Definitions: Energy conversion and efficiencies - System, property and state - Thermal equilibrium - Temperature - Zeroth law of Thermodynamics – P, V, and T Diagrams, - Thermodynamic Diagram. **(12)**

UNIT – II

First Law of Thermodynamics: The concept of work and adiabatic process - First law of thermodynamics - Conservation of Energy principle for closed and open systems - Calculation of work for different processes of expansion of gases.

(12)

UNIT – III

Second Law of Thermodynamics: Equilibrium and the second law - Heat engines - Kelvin-Planck statement of second law of thermodynamics - Reversible and irreversible processes - Carnot principle - Clausius inequality- Entropy. **(12)**

UNIT – IV

Gas Power Cycles: Air standard cycles: The air standard Carnot cycle - Air standard Otto cycle, diesel cycle, dual cycle and Bryton cycles and their efficiencies. **(12)**

UNIT – V

Refrigeration Cycles and Systems: Reverse Carnot cycle - COP - Vapor compression refrigeration cycle and systems (only theory) - Gas refrigeration cycle - Absorption refrigeration system - Liquifaction and solidification (only theory) **(12)**

Text Books:

1. Nag, P. K., "Engineering Thermodynamics", 4th edition, Tata Mc Graw Hill Publishing Co. Ltd., New Delhi,1995

Reference Books:

1. Arora, C.P., "Thermodynamics", Tata Mc Graw Hill Publishing Co. Ltd., New Delhi,1998.
2. Burghardt, M.D., "Engineering Thermodynamics with Applications", 4th edition, Harper & Row, N.Y., 1986.
3. Huang, F.F., "Engineering Thermodynamics" 2nd edition , Macmillan Publishing Co. Ltd., N.Y.,1989.
4. Cengel, Y.A. and Boles, M.A., "Thermodynamics - An Engineering Approach", 5thedition, Mc-Graw Hill, 2006
5. Wark, K., "Thermodynamics", 4th edition ,Mc Graw Hill, N.Y.,1985

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

Kalitheerthalkuppam, Madagadipet, Puducherry - 605 107

DEPARTMENT OF GENERAL ENGINEERING

REPORT

ON

DEMONSTRATION OF PETROL & DIESEL ENGINES WORKING

18th February 2021

Submitted by

Mr. A.THIAGARAJAN

Assistant Professor

Dept. of Mechanical Engienering

Manakula Vinayagar Institute of Technology

Puducherry

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY, KALITHEERTHALKUPPAM, PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

Kalitheerthalkuppam, Madagadipet, Puducherry - 605 107

Department of General Engineering

Circular

No: MIT/GE/Circular/2020-21/ **Date: 16/02/2021**

A Thermal Engineering Lab visit in our college is planned for first year Information Technology and Mechanical Engineering students to recognize the main components of Internal Combustion Engine and its working principle. The visit is scheduled on **18th February 2021**. All the students are informed to be present for the lab visit and make use of it.

16.2

Staff In-charge

16/2/21

Coordinator

Copy to:

All First year classes

Circulates to General Engineering staff Members

File.

PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

Kalitheerthalkuppam, Madagadipet, Puducherry - 605 107

1. INTRODUCTION

A Lab visit to Thermal Engineering lab was arranged on 18h February 2021. First year engineering students joined the visit under the guidance of Mr.A.THIAGARAJAN Asst. Professor, Dept of Mechanical Engineering. He explained well about the components and working of the petrol and diesel Engines. Students observed and understood the working principles of the Engines. The functions within an internal combustion engine was explained. The visit becomes useful for students development and has increased the span of students' knowledge.

2. OBJECTIVE OF THE VISIT

The Objective of visit was, students are able

- To understand the working of petrol and diesel Engines
- To identify the correct air and fuel ratio for an internal combustion engine.

3. PURPOSE OF VISIT.

Main purpose for this visit was to give the practical knowledge about the working of petrol and diesel Engines .To know the main difference in functioning of petrol and diesel Engines

4. OUTCOME OF THE FIELD VISIT

Students will be able:

1. Students understood the combustion processes and related components with their functions.
2. Students learned various operating conditions and parameters of correct air and fuel ratio for an internal combustion engine

5. CONCLUSION

Through this visit, students gained information and practical knowledge about various components used in petrol and diesel Engines. They have learnt about construction and working of various combustion engine components and their functions.

18.2.21

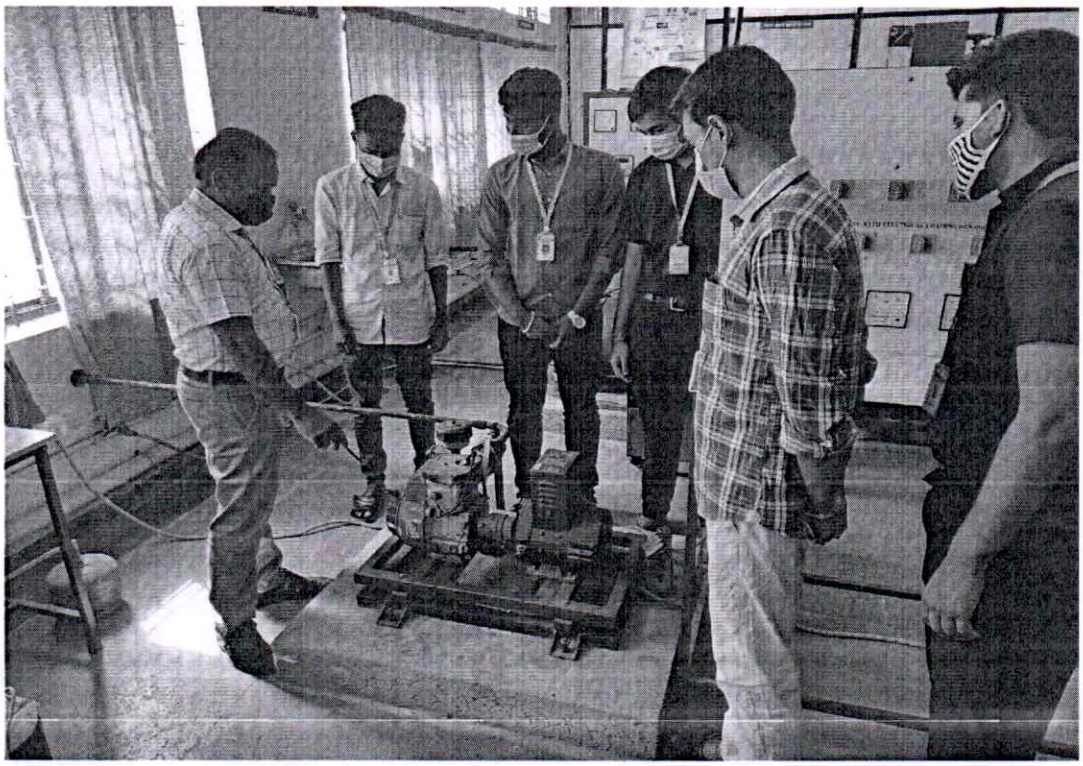
STAFF - INCHARGE

COORDINATOR

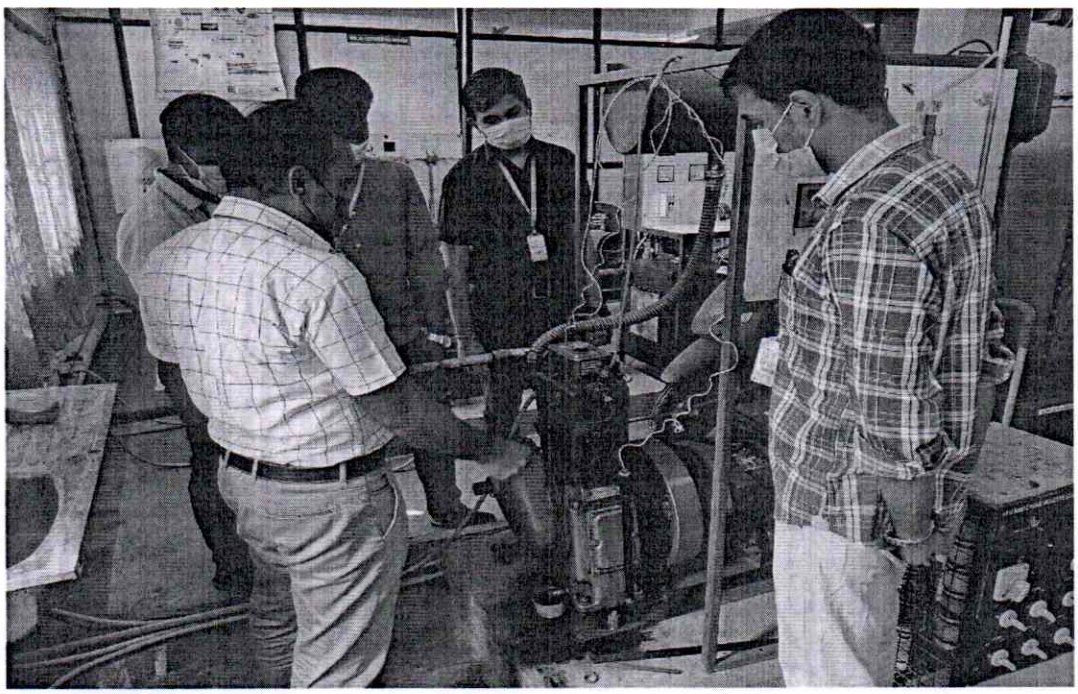
PUDUCHERRY

PRINCIPAL

PHOTOS



First year students learning about different components of Petrol Engines



Faculty explains the working principles of Petrol Engines

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

Kalitheerthalkuppam, Madagadipet, Puducherry - 605 107

DEPARTMENT OF GENERAL ENGINEERING
THERMAL LAB VISIT – WORKING OF PETROL & DIESEL ENGINES

ATTENDANCE ON 18/02/2021

Sl. No	Name of the Students	Signature
1	ABINAYA.R	
2	COUMMARESH A	
3	HARINI.I	
4	HARISHANKAR.B	
5	HEMA.R	
6	JENIFER.H	
7	JHON LEO .E	
8	KUMARAVEL.A.V	
9	LIVIYA.P	
10	LOGESHWAR.M	
11	MAHADEVI.J	
12	MAHESWARI.G	
13	MOHAMED HASIM B	
14	MUHAMED ANWARDHEEN.A	
15	MUTHAMIZHARASI .G	
16	NANDHAKUMAR .P	
17	PARTHIBAN.R	
18	PAVITHRA .K	
19	PRIYADHARSHINI.S	
20	REVANTH N	
21	SANGEETHA.R	
22	SANJAY.S	
23	SANTHOSHKUMAR.S	
24	SHAMELI.S	
25	SHEIK IMRAN.A	
26	SHYAM SUNDAR.H	
27	SRINIVASAN.L	
28	SRINIVASAN.P	

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY

29	SRIRANJAANI.G	
30	SUBAGAYATHRI.S	
31	THAMIZHSELVI.R	
32	THARUN.G	
33	UDHAYA SANKAR.S	
34	VENGADESAN.S	
35	VENKATESH.S	
36	VIGNESH.M	
37	VISHWA.D	
	MECHANICAL	
1	ABDUE FAHAD	
2	BALAMURUGAN.K	
3	DHANVANTHIRI RAJA R	
4	DIWAGAR.R	
5	HARSHATHKUMAR.P	
6	HIDHESSH.R	
7	MOHAMMED IMTHIYAS.A	
8	MURUGAN.S	
9	NIVASH.K	
10	RAMGOPAL .M	
11	RANJITH.R	
12	SACHIEL AROKUIAM. A	
13	SANJAI.S	
14	SILAMBARASAN	
15	SINDHAN.V.R	
16	YOHARAJESH.S	
17	YOUVARAJ.M	

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

T 102 – PHYSICS

COURSE OBJECTIVE

- *To understand the concepts of physics and its significant contributions in the advancement of technology and invention of new products that dramatically transformed modern-day society.*
- *To expose the students to different areas of physics which have direct relevance and applications to different Engineering disciplines*
- *To understand the concepts and applications of Ultrasonics, optics and some optical devices, Lasers and Fiber optics, Nuclear energy sources and wave mechanics*

COURSE OUTCOME

On successful completion of the module students will be able to:

- *Apply knowledge of science and engineering to understand physics and its significant contributions in the advancement of technology and invention of new products that dramatically transform modern-day society.*
- *Identify different areas of physics which have direct relevance and applications to different Engineering disciplines.*
- *Apply fundamental knowledge to understand applications of Ultrasonics, optics and some optical devices, Lasers and Fiber optics, Nuclear energy sources and wave mechanics.*

UNIT – I

Acoustics & NDT: Ultrasonics- Ultrasonic Waves Productions (Piezoelectric & Magnetostriction method) – Detections (Acoustic Grating). NDT application – Ultrasonic Echo method – Liquid Penetrant method

Acoustics - Factors affecting Acoustic of Buildings (Reverberation, Loudness, Focusing, Echo, Echelon Effect and Resonance) and their Remedies - Sabine's formula for Reverberation Time–Doppler effect and its application to Random's (elementary idea) **(12)**

UNIT – II

Optics: Interference- Air Wedge – Michelson's Interferometer – Wavelength Determination – Interference Filter – Antireflection Coatings.

Diffraction - Diffraction Grating – Dispersive power of grating - Resolving Power of Grating & Prism Polarization –Basic concepts of Double Refraction - Huygens Theory of Double Refraction – Quarter and Half Wave Plates – Specific Rotary Power – Laurent Half Shade Polari meter. **(12)**

UNIT – III

Lasers & Fiber Optics: Lasers - Principles of Laser – Spontaneous and Stimulated Emissions - Einstein's Coefficients – Population Inversion and Laser Action – Types of optical resonators(qualitative Ideas) – Types of Lasers - NdYAG, CO2 laser, GaAs Laser – Application of Lasers.

Fiber Optics - Principle and Propagation of light in optical fiber – Numerical aperture and acceptance angle – Types of optical fibers (material, refractive index, mode)- Application to sensors and Fiber Optic communication. **(12)**

UNIT – IV

Wave Mechanics: Matter Waves – de Broglie Wavelength – Uncertainty Principle – Schrödinger Wave Equation – Time Dependent – Time Independent – Application to Particle in a One Dimensional potential Box – Quantum Mechanical Tunneling – Tunnel Diode. **(12)**

UNIT – V

Nuclear energy source: General Properties of Nucleus (Size, Mass, Density, Charge) – Mass Defect – Binding Energy - Disintegration in fission – Nuclear Reactor: Materials Used in Nuclear Reactors. – PWR – BWR – FBTR. Nuclear fusion reactions for fusion reactor – D-D and D-T reactions, Basic principles of nuclear fusion reactors. **(12)**

Text Books

1. V Rajendran, Engineering Physics, 2nd Ed., TMH, New Delhi 2011 (For Units I to IV only)
2. Arthur Beiser, Concept of Modern Physics, 6th Ed, TMH, New Delhi 2008 (For Unit V Only)

Reference Books

1. Ajay Ghatak, Optics, TMH, New Delhi, 2007.
2. Thiagarajan and Ghatak, Laser and Application, TMH, New Delhi 2008.
3. R. Murugeshan, Modern Physics, S. Chand & Co, New Delhi 2006
4. K.R.Nambiar, Lasers, New Age International, New Delhi, 2008
5. Science of Engineering Materials, 2nd Edition, C.M. Srivastava and C. Srinivasan, New Age Int. (P) Ltd, New Delhi, 1997
6. Avadhanulu M N Engineering Physics, Vol-I, S. Chand & Co,2009.

T 108 - MATERIAL SCIENCE

COURSE OBJECTIVE

- *To understand the importance of Material Science as a subject that revolutionized modern day technologies*
- *To understand the significance of material science in the development of new materials and devices for all branches of Engineering*
- *To impart knowledge to the Engineering students about some of the important areas of Materials Science so as to enable them perceive the significant contributions of the subject in Engineering and Technology*

COURSE OUTCOME

On successful completion of the module students will be able to:

- *Apply knowledge of mathematics, science and engineering to understand the importance of Material Science as a subject that revolutionized modern day technologies*
- *To analyze the significance of material science in the development of new materials and devices for all branches of Engineering*
- *Identify and analyze some of the important areas of Materials Science so as to enable them perceive the significant contributions of the subject in Engineering and Technology*

UNIT – I

Crystal structure and Defects: Crystal structure - Bravais Lattices , Crystal Systems - Coordination Number, Atomic Radius, Packing Factor for FCC & HCP structures – Miller Indices- Powder X Ray Diffraction Method Lattice defects – Qualitative ideas of point, line, surface and volume defects. **(12)**

UNIT – II

Dielectric properties: Dielectric Polarization and Mechanism –Temperature dependence of polarization, Internal or local Field - Clausius-Mossotti relation. Basic ideas of Dielectric loss - frequency dependence of dielectric constant – Measurement of Dielectric constant and loss using Scherring bridge – Elementary ideas of Piezoelectrics, Ferroelectrics and Pyroelectric materials and Applications. **(12)**

UNIT – III

Magnetic Properties: Origin of atomic magnetic moment – Bohr magneton-Elementary Ideas of classification of magnetic materials (Dia, Para, Ferro, antiferro & Ferri). – Quantum theory of Para & Ferro Magnetism – Domain Theory of Hysteresis –

Heisenberg Theory of Exchange Interaction (without derivation) – Qualitative ideas of Anti ferromagnetic Ordering – Structure and Properties of Ferrites – Properties of Soft & Hard Magnetic Materials – Applications. Magnetic data storage – Magnetic tapes, Hard disks, Magneto optical recording. **(12)**

UNIT – IV

Semiconductors and superconductors: Semiconductors -Derivation of Carrier concentration in intrinsic Semiconductors –Basic ideas of Electrical conductivity in intrinsic and extrinsic semiconductors (without derivations) -temperature dependence of carrier concentration and electrical conductivity in semiconductors (qualitative ideas), Hall effect in Semiconductors --Application of Hall Effect, Basic Ideas of Compound Semiconductors (II-VI & III-V)

Superconductivity - Basic concepts – transition temperature – Meissener effect – Type I and II superconductors – High Temperature Superconductors – 123 superconductor – Applications of superconductors. **(12)**

UNIT – V

Advanced Materials: Liquid Crystals – Types – Application as Display Devices

Metallic Glasses – preparation by melt spinning. Twin roller system, properties and applications

Shape Memory alloys (SMA), Shape memory effect, Properties and applications of SMA

Nanomaterials - Nano materials (one, Two& three Dimensional) –Methods of synthesis (PVD, CVD, Laser Ablation, Solgel, Ball-milling Techniques), Properties and applications of nanomaterials. carbon nanotubes – synthesis, Properties and applications. **(12)**

Text Books:

1. V Rajendran, Engineering Physics, 2nd Edition, TMH, New Delhi 2011.

Reference Books:

1. Ali Omar M, Elementary Solid State Physics, Addison Wesley Publishing Co., 2009.
2. William D Callister Jr., Material Science and Engineering, 6th Edition, John Wiley and sons, 2009.
3. Charles Kittel, Introduction to Solid State Physics, 7th Edition, John Wiley & sons, Singapore, 2007.
4. V Raghavan , Materials Science and Engineering- A First Course, 5th Edition, Prentice Hall of India, 2008.

5. B.S. Murty, P. Shankar, Baldev Raj, B.B. Rath, and James Murday, Text book of Nanoscience and Nanotechnology, Universities Press, Hyderabad 2012
6. M.N. Avadhanulu, Enginerring Physics- Volume-II, S.Chand &Co, New Delhi, 2009
7. Pillai S.O, Solid State Physics, 6th Edition – New Age International, 2005.

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T32	Electronic Devices and Circuits	3	1	0

Course Objectives:

1. To introduce the applications of PN junction diode and Zener diode
2. To familiarize the students with an in-depth knowledge of special devices
3. To impart knowledge on biasing of BJT and FET.
4. To introduce the construction and operation of oscillators.
5. To introduce the op-amp fundamentals and to teach the applications of op-amp

Course Outcomes:

On successful completion of the module students will be able to:

1. Analyze the behavior of PN junction diode, Zener diode and other special devices.
2. Understand the application areas of diodes.
3. Gain knowledge in biasing of BJT, FET.
4. Understand the working of Power amplifiers and oscillators.
5. Understand the practical applications of op-amps.

Unit I-Diode and Its Applications

Diode current equation – V-I characteristics of PN junction diode – Applications- Half wave and Full wave rectifiers with and without filters, Derivation of ripple factors ,rectification efficiency and transformer utilization factor, Zener diode and its application, Clippers, Clampers, voltage multipliers.

Unit II-Special Diodes

Special devices: Silicon controlled rectifier, Uni-junction transistor, LED, LCD, Schottky Barrier diode, Varactor diode, Tunnel diode , photodiode, photo-transistor

Unit III-Small Signal Amplifiers

BJT – Transistor biasing and bias circuits – operating point — 'h' parameters — FET biasing – Power Amplifier – Types – Transformer coupled Class A Amplifier – Class B Amplifier – Amplifier distortion- Class C and Class D amplifiers.

Unit IV-Feedback Amplifiers

Feedback concept, general characteristics of positive feedback, Oscillators: Barkhausen Criterion- Hartley, Colpitts, Wien bridge oscillators and crystal oscillator – frequency stability.

Unit V-Operational Amplifier

Introduction to op-amp, Characteristics of op-amp, Op-amp parameters - Equivalent circuit - Applications: Inverting and non-inverting amplifier, summer, subtractor, voltage follower, differentiator, integrator, comparator, first order low pass and high pass active filters.

(Total: 60 Periods)

Content beyond Syllabus:

PSPICE Simulation on device characteristics (optional)

Text Books:

1. Robert L.Boylestad and Louis Neshelsky, Electronic devices and circuit theory, 11th Edition, Prentice Hall India, 2012.
2. Jacob Millman and Arvin Grabel, Micro-Electronics, McGraw Hill, Fifth edition, 2008.

Reference Books:

1. Jacob Millman and C. Halkias, Satya brataJit, Electronic Devices and circuits, Second edition, McGraw Hill Publications, 2007.
2. Theodore F.Bogart and etal, Electronic Devices and Circuits, pearson Education, 2004

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF IT

Mini Project on VOLTAGE REGULATOR USING LM317

Submitted by

ABINESH.S
AISWARIYA.S
ANITHA.K
ARAVINDH.M
BHARATHI.P

II YEAR IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

VOLTAGE REGULATOR USING LM317

Aim:

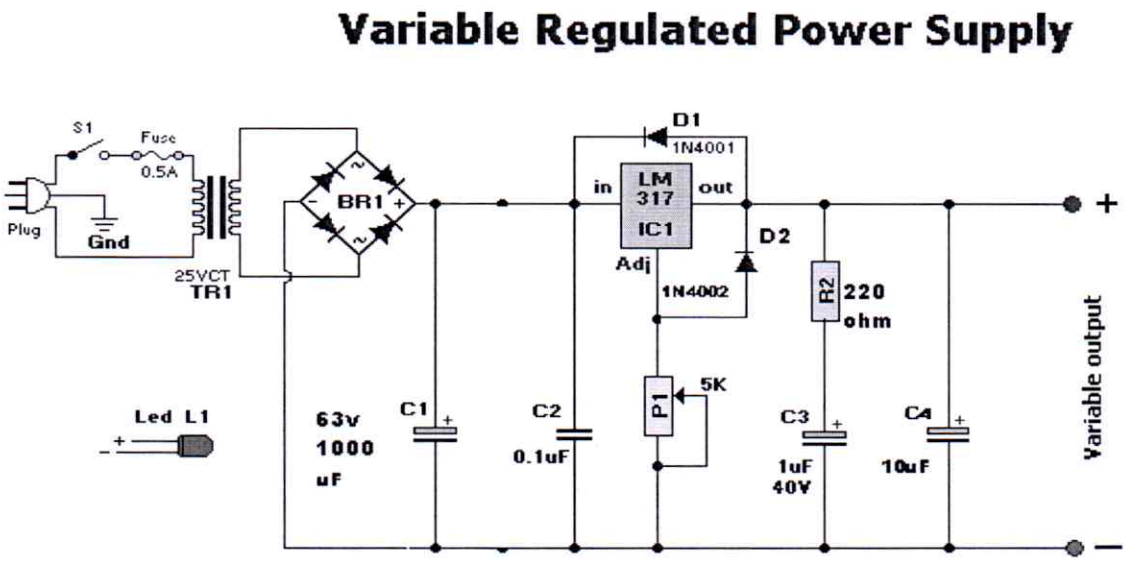
To Study of LM317 serves a wide variety of applications including **local, on card regulation.** This device can also be used to make a programmable output regulator, or by connecting a fixed resistor between the adjustment and output, the LM317 can be used as a precision current regulator.

Components Required:

SPECIFICATIONS

BR1 = BRIDGE RECTIFIER, 100V - 3A
C1 = 1000 µF, 63V
IC1 = LM317, ADJUSTABLE REGULATOR
C2 = 0.1 µF

Circuit Diagram:



PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
Puducherry
605 107

IT-T33 DATA STRUCTURES

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T33	Data Structures	3	1	0

Course Objectives:

To introduce the primary data structures and the associated operations
To understand the applications of data structures with case studies
To learn the implementation issues of the data structures introduced

Course Outcomes:

On successful completion of this course students will be able to:

Use appropriate data structures in programming
Learn various ways of implementing the data structures

Unit I **(12 Periods)**

Basics : Abstract Data Type(ADT) – introduction to data structures – representation - implementation

Stack and list: representing stack – implementation – application – balancing symbols –conversion of infix to postfix expression – evaluating a postfix expression – recursive function call – Linked list ADT – implementation using arrays – limitations - linked list using dynamic variables- linked implementation of stacks – circular list – doubly linked lists

Unit II **(12 Periods)**

Queues: Queue abstract data type - Array implementation – circular queue - linked list implementation of queues – priority queues – double ended queues – multiple stacks and queues - application.

Unit III **(12 Periods)**

Trees : General trees – binary tree – traversal methods – expression trees – game trees. Binary search trees – AVL trees – Splay trees – B Trees – B^+ Trees – Tries – application.

Unit IV **(12 Periods)**

Sorting: O notation – efficiency of sorting – bubble sort – quick sort – selection sort – heap sort – insertion sort – shell sort – merge sort – radix sort.

Unit V **(12 Periods)**

Hashing: Introduction – Hash function – methods - Hash table implementation - rehashing.

Graph: Directed and un directed graph – representation of graphs – graph traversals: Depth first search – Breadth first search – transitive closure – spanning trees – application - topological sorting.

(Total: 60 Periods)

Content beyond Syllabus:

1. Advanced data structures and their implementation.
2. Implementation of the data structures in different language platforms.

Text Books:

1. Mark Allen Weiss, Data structures and algorithm analysis in C++, Pearson Education, 6th edition, 2011
2. YedidyahLangsam, Moshe J Augenstein and Aaron M Tanenbaum, Data Structures using C and C++, 2nd edition, Prentice Hall of India, 2009.

Reference Books:

1. G.A.V.Pai, Data Structures and Algorithms – Concepts, Techniques and Applications, Tata McGraw Hill Publishing Company Limited, New Delhi, 2008.
2. Ellis Horowitz and SartajSahni, Fundamentals of Data structures, Galgotia Publications, 2nd Edition, New Delhi, 2001.
3. Alfred V. Aho, Jeffrey D. Ullman, John E. Hopcroft. Data Structures and Algorithms. Addison Wesley, 1983

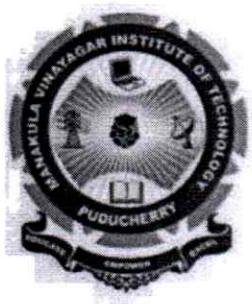
Websites:

http://www.cs.sunysb.edu/~skiena/214/lectures/
http://opendatastructures.org/
http://www.cplusplus.com/doc/tutorial/structures/

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY



IT – T33 DATA STRUCTURE

Mini Project on

HASH TABLE USING OPEN HASHING OR SEPARATE TECHNIQUE & SIMPLE GRAPH USING THE ADJACENCY LIST

Submitted by

GANESH .T
GURUPRASAD. G
HEMAVATHI J
JAYA PRIYA.S
KALAIVANI. M
KALYANI VARSHINI. B
KEERTHANA.S

II YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY 605 107

HASH TABLE USING OPEN HASHING OR SEPARATE TECHNIQUE

AIM:

To implement the hash table using the open hashing or Separate technique

DESCRIPTION:

In the below code, an array is created for all the keys that need to be inserted in the has table. Class and constructors are created for hash MapTable to calculate the hash function using the formula mentioned above. The list is created as the pointer to the array of key values. Specific functions are created for the insertion, deletion, and display of the hash table and called from the main method. While insertion, if 2 or more elements have the same index, they are inserted using the list one after the other.

ALGORITHM:

Step 1: Start the program
Step 2: Create class as HashMapTable and declare size of the hash table.
Step 3: Create constructor of the class containing all the methods.
Step 4: Declare the array size and value which to be inserted into the hash table.
Step 5: Insert the key in the hash table
Step 6: Delete the key in the hash table
Step 7: Display the hash table using displayHashTable()
Step 8:Stop the program.

SOURCE CODE:

```
#include <iostream>
#include <list>
using namespace std;
class HashMapTable
{
// size of the hash table
inttable_size;
// Pointer to an array containing the keys
list<int> *table;
public:
// creating constructor of the above class containing all the methods
HashMapTable(int key);
// hash function to compute the index using table_size and key
inthashFunction(int key) {
return (key % table_size);
}
// inserting the key in the hash table
void insertElement(int key);
// deleting the key in the hash table
void deleteElement(int key);
// displaying the full hash table
```

SIMPLE GRAPH USING THE ADJACENCY LIST

AIM:

To write a C++ program to implement and demonstrate a simple graph using the adjacency list.

DESCRIPTION:

Here we are going to display the adjacency list for a weighted directed graph. We have used two structures to hold the adjacency list and edges of the graph. The adjacency list is displayed as (start_vertex, end_vertex, weight).

ALGORITHM:

Step1: Start the program.
Step2:Create the structure to hold the adjacency and edges of the graph.
Step 3: Create a class DiaGraph to insert new nodes into adjacency list from given graph.
Step 4: Insert the new nodes into the graph.
Step 5: Display the all adjacent vertices of given vertex.
Step 6:Stop the program

SOURCE CODE:

```
#include <iostream>
using namespace std;
// stores adjacency list items
struct adjNode {
   int val, cost;
   adjNode* next;
};
// structure to store edges
struct graphEdge {
   int start_ver, end_ver, weight;
};
class DiaGraph{
   // insert new nodes into adjacency list from given graph
   adjNode* getAdjListNode(int value, int weight, adjNode* head)   {
      adjNode* newNode = new adjNode;
      newNode->val = value;
      newNode->cost = weight;

      newNode->next = head;   // point new node to current head
      return newNode;
   }
   int N;  // number of nodes in the graph
public:
```

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – T33 DATA STRUCTURE

Mini Project on

DOUBLE ENDED QUEUE & DEPTH FIRST SEARCH TRAVERSAL

Submitted by

LAVANYA.S
LOGANATHAN .V
MAHADEVAN. R
MALINI.P
MOHAMED AKHIL.R
MOHAMED FAGAD.I

II YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY 605 107

DOUBLE ENDED QUEUE

AIM:

To write a C++ program to implement dequeue (Double ended queue).

DESCRIPTION:

Dequeue or Double Ended Queue is a generalized version of Queue data structure that allows insert and delete at both ends.

Some basic operations of dequeue are –

insert_at_beg(): inserts an item at the front of Dequeue.

insert_at_end(): inserts an item at the rear of Dequeue.

delete_fr_beg(): Deletes an item from front of Dequeue.

delete_fr_rear(): Deletes an item from rear of Dequeue.

ALGORITHM:

Begin

Declare a class dequeue to declare front f and rear r and following functions:

function insert_at_beg(int) to insert item at front:

If queue is not completely filled up, insert element at the front and update front and rear

Otherwise print overflow.

function insert_at_end(int) to insert item at rear:

If queue is not completely filled up, insert element at the rear and update front and rear

Otherwise print overflow.

function delete_fr_beg() to delete item from front:

If queue is empty, print underflow otherwise delete the front element and update front.

function delete_fr_end() to delete item from end:

If queue is empty, print underflow otherwise delete the rear element and update rear.

End

DEPTH FIRST SEARCH TRAVERSAL

AIM:

To implement the DFS (Depth First Search) traversal technique using C++.

DESCRIPTION:

Unlike BFS in which we explore the nodes breadthwise, in DFS we explore the nodes depth-wise. In DFS we use a stack data structure for storing the nodes being explored. The edges that lead us to unexplored nodes are called 'discovery edges' while the edges leading to already visited nodes are called 'block edges'.

ALGORITHM:

Step 1: Insert the root node or starting node of a tree or a graph in the stack.
Step 2: Pop the top item from the stack and add it to the visited list.
Step 3: Find all the adjacent nodes of the node marked visited and add the ones that are not yet visited, to the stack.
Step 4: Repeat steps 2 and 3 until the stack is empty.\

SOURCE CODE:

```
#include <iostream>
#include <list>
using namespace std;
//graph class for DFS travesal
class DFSGraph
{
int V;   // No. of vertices
list<int> *adjList;   // adjacency list
void DFS_util(int v, bool visited[]);  // A function used by DFS
public:
   // class Constructor
DFSGraph(int V)
   {
 this->V = V;
 adjList = new list<int>[V];
   }
   // function to add an edge to graph
void addEdge(int v, int w){
adjList[v].push_back(w); // Add w to v's list.
   }

void DFS();   // DFS traversal function
};
void DFSGraph::DFS_util(int v, bool visited[])
{
   // current node v is visited
```

T 106 - COMPUTER PROGRAMMING

COURSE OBJECTIVE

- *To introduce the basics of computers and information technology and educate problem solving techniques.*
- *To impart programming skills in C language.*
- *To practice structured programming to solve real life problems.*

COURSE OUTCOME

On successful completion of the module students will be able to:

- *Apply fundamental knowledge of science and engineering to understand the basics of computers and information technology and educate problem solving techniques.*
- *Apply logical thinking to create programs in C language.*
- *Design system, component and demonstrate structured programming to solve real life problems*

UNIT – I

History of Computers – Block diagram of a Computer – Components of a Computer system – Classification of computers - Hardware – Software – categories of Software – Operating System – Applications of Computers –Network Structure - Internet and its services – Intranet – Study of word processor – Preparation of worksheets. **(12)**

UNIT – II

Problem solving techniques – Program – Program development cycle – Algorithm design – Flowchart - Pseudo code. Introduction to C – History of C – Importance of C - C tokens – data types – Operators and expressions – I/O functions. **(12)**

UNIT – III

Decision making statements – branching and looping – arrays – multidimensional arrays – Functions – Recursion – Passing array to functions .Storage classes – Strings – String library functions. **(12)**

UNIT – IV

Structures – Arrays and Structures – nested structures – passing structures to functions – user defined data types– Union. Pointers – pointers and arrays – pointers and functions - pointers and strings - pointers and structures. **(12)**

UNIT – V

Files – operations on a file – Random access to files – command line arguments .Introduction to preprocessor – Macro substitution directives – File inclusion directives – conditional compilation directives – Miscellaneous directives. **(12)**

Text Books:

1. Balagurusamy. E, "Programming in ANSI C", Tata McGraw Hill, 12th Edition, 2012

Reference Books:

1. Vikas Verma, "A Workbook on C ",Cengage Learning, Second Edition,2012
2. Ashok N Kamthane, "Computer Programming", Pearson education, Second Impression, 2008.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM,
PUDUCHERRY-605 107.

IT-T34 OBJECT ORIENTED PROGRAMMING

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T34	Object Oriented Programming	3	1	0

Pre-requisite: Structured Programming Language C

Course Objectives:

1. To understand the concepts of object-oriented programming and master OOP using C++.

Course Outcomes:

On successful completion of this course students will be able to:

1. Analyze and design a problem using an object-oriented approach.
2. Implement the problem using C++ programming Language.

Unit I

Object oriented programming - concepts – objects – classes – methods and messages – abstraction and encapsulation – inheritance – abstract classes – polymorphism. Introduction to C++ – classes – access specifiers – function and data members – default arguments – function overloading – friend functions – const and volatile functions - static members – Objects – pointers and objects – constant objects – nested classes – local classes

Unit II

Constructors – default constructor – Parameterized constructors – Constructor with dynamic allocation – copy constructor – destructors – operator overloading – overloading through friend functions – overloading the assignment operator – type conversion – explicit constructor

Unit III

Function and class templates - Exception handling – try-catch-throw paradigm – exception specification – terminate and Unexpected functions – Uncaught exception.

Unit IV

Inheritance – public, private, and protected derivations – multiple inheritance - virtual base class – abstract class – composite objects Runtime polymorphism – virtual functions – pure virtual functions – RTTI – typeid – dynamic casting – RTTI and templates – cross casting – down casting .

Unit V

Streams and formatted I/O – I/O manipulators - file handling – random access – object serialization – namespaces - std namespace – ANSI String Objects – standard template library.

(Total : 60 Periods)

Content beyond Syllabus:

1. Implementation of the design patterns to the solution of programming problems.

Text Books:

1. B.Trivedi, "Programming with ANSI C++", Oxford University Press, 2007.

Reference Books:

1. Ira Pohl, "Object Oriented Programming using C++", Pearson Education, Second Edition Reprint 2004..
2. S. B. Lippman, JoseeLajoie, Barbara E. Moo, "C++ Primer", Fourth Edition, Pearson Education, 2005.
3. B. Stroustrup, "The C++ Programming language", Third edition, Pearson Education,2004.
4. D. S. Malik, C++ Programming: From Problem Analysis to Program Design, 2012
5. E. Balaguruswamy,ObjectOriented Programming with C++,6th edition, TMH,2013.

Websites:

1. http://www.cplusplus.com

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-T34 OBJECT ORIENTED PROGRAMMING

Mini Project on Bank Management System

Submitted by

MOHAMED THALHAA.F

MOUNISHAA.P

NANTHINI.N

NIVAAS.D

PATHRINATH M

PRAVEEN.V

II YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

Bank Management System

Objective:

In this C++ Projects we are going to **create C++ Bank Management System**. This application which can be used by Customers for Bank Related Transactions. It is a very simplified application and is very user friendly. We developed this project under Codeblocks editor. This Bank Management System project consist of below modules

- Create New Bank Account
- Deposit Amount to the Account
- Withdraw amount from your account
- Check Balance Inquiry
- Display All Account Holders List
- Update Bank Account
- Close Account

PROGRAM:

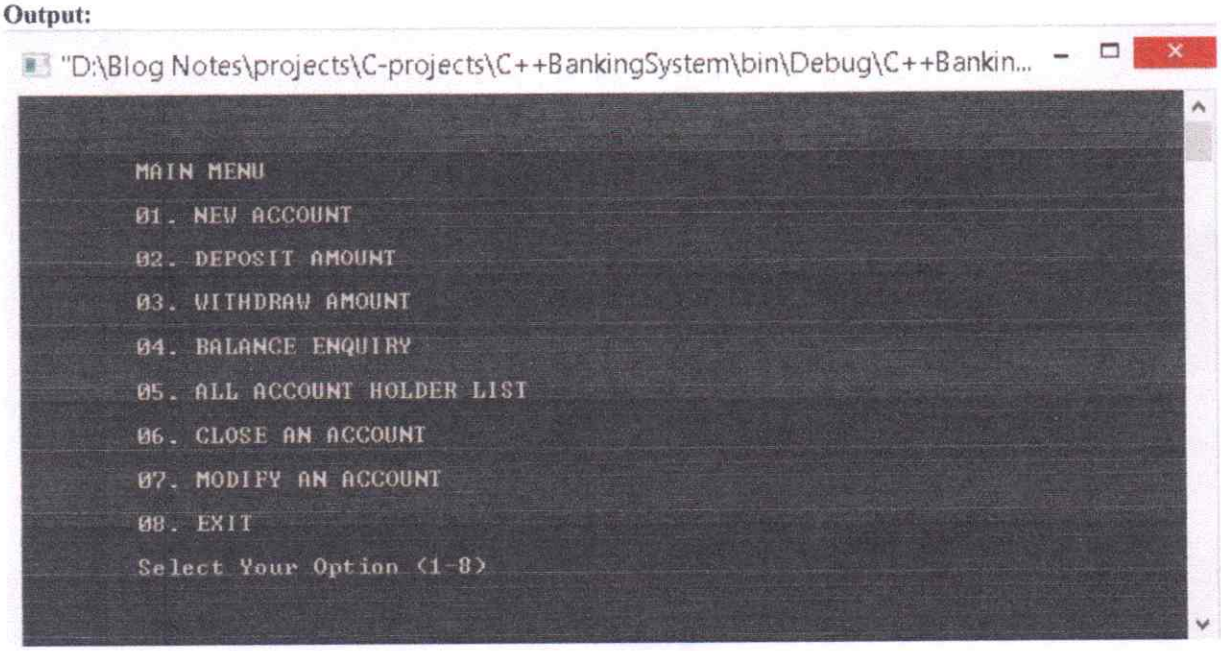
```
#include <iostream>
#include<iostream>
#include<fstream>
#include<cctype>
#include<iomanip>
using namespace std;

void write_account();  //function to write record in binary file
void display_sp(int);  //function to display account details given by user
void modify_account(int);  //function to modify record of file
void delete_account(int);  //function to delete record of file
void display_all();      //function to display all account details
void deposit_withdraw(int, int); // function to desposit/withdraw amount for given account
void intro();  //introductory screen function
int main()
{
  char ch;
  int num;
  intro();
  do
  {
    system("cls");
    cout<<"\n\n\n\tMAIN MENU";
    cout<<"\n\n\t01. NEW ACCOUNT";
    cout<<"\n\n\t02. DEPOSIT AMOUNT";
    cout<<"\n\n\t03. WITHDRAW AMOUNT";
    cout<<"\n\n\t04. BALANCE ENQUIRY";
    cout<<"\n\n\t05. ALL ACCOUNT HOLDER LIST";
    cout<<"\n\n\t06. CLOSE AN ACCOUNT";
    cout<<"\n\n\t07. MODIFY AN ACCOUNT";
    cout<<"\n\n\t08. EXIT";
    cout<<"\n\n\tSelect Your Option (1-8) ";
    cin>>ch;
    system("cls");
    switch(ch)
    {
    case '1':
                ;
      break;
    case '2':
```

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

```
    }
  File.close();
  if(found==false)
    cout<<"\n\n Record Not Found ";
}


//***************************************************************
//    INTRODUCTION FUNCTION
//***************************************************************


void intro()
{
  cout<<"\n\n\n\t  BANK";
  cout<<"\n\n\tMANAGEMENT";
  cout<<"\n\n\t  SYSTEM";
  cout<<"\n\n\n\nMADE BY : RRTutors";
  cout<<"\n\n Bank : Bank Of RRT";
  cin.get();
}
```

Output:



Conclusion: In c++ we create bank management System Project with Add account, display balance, modify account, close bank account features.

T101 - MATHEMATICS – I

COURSE OBJECTIVE

- *To introduce functions of several variables and the idea of applying calculus concepts to problems in Engineering.*
- *To acquaint the student with mathematical tools needed in evaluating multiple integrals and their usage.*
- *To introduce effective mathematical tools for the solutions of differential equations that model physical processes.*

COURSE OUTCOME

On successful completion of the module students will be able to:

- *Apply knowledge of mathematics to solve functions of several variables.*
- *Identify, formulate, and solve engineering problems likemultiple integrals and their usage.*
- *To solve differential equations that model physical processes using effective mathematical tools*

UNIT – I

Calculus: Curvature, radius of curvature, evolutes and involutes. Beta and Gamma functions and their properties. **(12)**

UNIT – II

Function of Several variables: Partial derivatives, Total derivatives, Differentiation of implicit functions, Change of variables, Jacobians and their properties, Taylor's series for functions of two variables, Maxima and minima, Lagrange's method of undetermined multipliers. **(12)**

UNIT – III

Multiple Integrals and Applications: Multiple Integrals, change of order of integration and change of variables in double integrals (Cartesian to polar). Applications: Areas by double integration and volumes by triple integration (Cartesian and polar). **(12)**

UNIT – IV

Differential Equations: Exact equations, First order linear equations, Bernoulli's equation, orthogonal trajectories, growth, decay and geometrical applications: Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type. **(12)**

UNIT – V

Differential Equations (Higher order): Linear differential equations of higher order – with constant coefficients, the operator D, Euler's linear equation of higher order with variable coefficients - simultaneous linear differential equations, solution by variation of parameters method– simple applications to electric circuits. **(12)**

Text Books:

1. Venkataraman, M. K, Engineering Mathematics (First Year), Second Edition, The National Publishing Company, Chennai 2010 (For units I, III, IV, V)
2. Grewal B.S., Higher Engineering Mathematics, Khanna Publishers, New Delhi, 41stEdition, 2011. (For Unit II only)

Reference Books:

1. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
2. Kandasamy P. et al, Engineering Mathematics, Vol.1 & 2, S. Chand & Co., New Delhi.
3. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
4. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons, New Delhi, 8th Edition.
5. Bali, N. P, and Manish Goyal, A Text Book of Engineering Mathematics, Lakshmi Publications, New Delhi, 2007

T 107 - MATHEMATICS – II

COURSE OBJECTIVE

- *To develop the use of matrix algebra techniques for practical applications and to introduce the concepts of Curl, Divergence and integration of vectors in vector calculus which is needed for many application problems.*
- *To introduce Laplace transform which is a useful technique in solving many application problems and to solve differential and integral equations*
- *To acquaint the students with Fourier transform techniques used in wide variety of situations in which the functions used are not periodic.*

COURSE OUTCOME

On successful completion of the module students will be able to:

- *Apply knowledge of mathematics to solve matrix algebra techniques for practical applications and Curl, Divergence and integration of vectors in vector calculus for many application problems.*
- *Identify, formulate, and solve engineering problems like Laplace transform which is a useful technique in solving many application problems and to solve differential and integral equations*
- *Apply formulae and analyze problems of Fourier transform techniques*

UNIT – I

Matrices: Eigenvalues and Eigen vectors of a real matrix, Characteristic equation, Properties of Eigenvalues and Eigenvectors. Cayley-Hamilton Theorem, Diagonalization of matrices. Reduction of a quadratic form to canonical form by orthogonal transformation. Nature of quadratic forms **(12)**

UNIT – II

Vector Calculus: Gradient, divergence and curl, their properties and relations. Gauss divergence theorem and Stoke's theorem (without proof). Simple application problems **(12)**

UNIT – III

Laplace Transform: Definition, Transforms of elementary functions, properties. Transform of derivatives and integrals. Multiplication by t and division by t. Transform of unit step function, transform of periodic functions. Initial and final value theorems **(12)**

UNIT – IV

Applications of Laplace Transform: Methods for determining inverse Laplace Transforms, convolution theorem, Application to differential equations and integral equations. Evaluation of integrals by Laplace transforms. **(12)**

UNIT – V

Fourier Transform: Fourier Integral theorem (statement only), Fourier transform and its inverse, properties. Fourier sine and cosine transforms, their properties, convolution and Parseval's identity. **(12)**

Text Books:

1. Venkataraman M.K, Engineering Mathematics The National Publishing Company, Chennai, 2012.
2. Kandasamy P. et al, Engineering Mathematics, Vol.2 & 3, S. Chand & Co., New Delhi.

Reference Books:

1. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
2. Grewal B.S., Higher Engineering Mathematics, Khanna Publishers, New Delhi,41st Edition, 2011.
3. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
4. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons, New Delhi.
5. Bali N. and Goyal M., Advanced Engineering Mathematics, Lakshmi Publications Pvt. Ltd., New Delhi, 7th Edition, 2010

IT-T35 DIGITAL SYSTEM DESIGN

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T35	Digital System Design	3	1	-

Pre-requisite: Basic Electronics

Course Objectives:

- To apply knowledge of number systems, codes and Boolean algebra to the analysis and design of digital logic circuits.
- To identify, formulate, and solve engineering problems in the area of digital logic circuit design.
- To use the techniques, skills, and modern engineering tools such as logic works and VHDL, necessary for engineering practice.
- To design a digital system, components or process to meet desired needs within realistic constraints

Course Objectives:

- To apply knowledge of number systems, codes and Boolean algebra to the analysis and design of digital logic circuits.
- To identify, formulate, and solve engineering problems in the area of digital logic circuit design.
- To use the techniques, skills, and modern engineering tools such as logic works and VHDL, necessary for engineering practice.
- To design a digital system, components or process to meet desired needs within realistic constraints

UnitI–Number Systems and Boolean Algebra **(12 Periods)**

Revision of logic gates – Binary number systems and conversion-Binary arithmetic-Binary codes-Boolean algebra - Basic operations-Basic Theorems - Boolean functions – Canonical forms - Simplification of Boolean functions – Karnaugh maps- Tabulation method.

UnitII–Combinational Logic **(12 Periods)**

Adders – subtractors– code converters – binary parallel adder –decimal adder – magnitude comparator –encoders–decoders–multiplexers–demultiplexers-BinaryMultiplier–Parity generator and checker.

UnitIII–Sequential Logic I **(12 Periods)**

Sequentialcircuits:latches–flipflops–analysisofclockedsequentialcircuits–statereduction and assignments - RegistersandCounters:Registers–shiftregisters–ripplecounters–synchronouscounters–ringcounters–up/downcounters–moduluscounters.

UNITIV –Sequential Logic II **(12 Periods)**

Memory and Programmable Logic: Random Access Memory–memory decoding–error detection and correction–ReadOnlyMemory–ProgrammableLogicArrays–ProgrammableArrayLogic.AsynchronousSequentialLogic:Analysisprocedure–circuitswithLatches–Designprocedure–Reduction of state and Flow tables–Race-Free state assignment–Hazards.

UNITV-Introduction to Verilog Hardware Description Language **(12 Periods)**

Introduction–HDLforcombinationalcircuits–Sequentialcircuits–Registersandcounters–HDLdescriptionforbinarymultiplier.

(Total : 60 Periods)

Content beyond Syllabus:

Design of ALU and simple computer logic

TEXT BOOKS

1. M.MorrisMano,DigitalDesign,4th edition,Prentice-HallofIndiaPvt. Ltd.,2006.

Reference Books:

1. ThomasL.Floyd,R.P.Jain,DigitalFundamentals,10th edition,PearsonEducation, 2008.
2. LeachMalvino,DigitalPrinciplesandApplications,5th edition,TataMcGrawHill,2005.
3. CharlesH.Roth,Fundamentalsof LogicDesign,5th edition,ThomsonBrooks/cole.2003.
4. ThomasCBartee,ComputerArchitectureandLogicDesign,McGrawHill,Singapore,2002.
5. T.R.Padmanabhan,DesignthroughVerilogHDL,Wiley-IEEEPress,2003.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

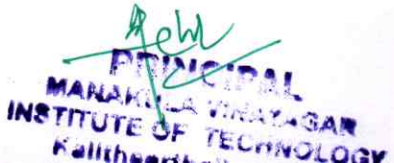
DEPARTMENT OF INFORMATION TECHNOLOGY

IT T35 Digital System Design

Mini Project on Design and testing of Ring Counter/Johnson Counter.

Submitted by

VASANTH ABIMANU K
VEL SENABATHY@VISHNU.SR
VERONICA T, ANUSUYA.R
VIDYA LAKSHMI. B
YUVARAJ.S, AASHIFA.S
KAMESH.M, AAKASH.G
AL BISEL.A, ARUN.D
ANNUNCIAMARY
DHANUJA SRI .P
II YEAR – IT
(2020-2021)



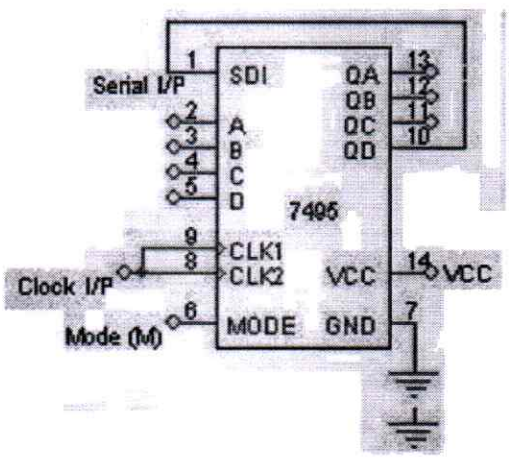
Aim: Design and testing of Ring Counter/Johnson Counter.

Apparatus Required: IC7495, IC7404, etc.

Procedure:

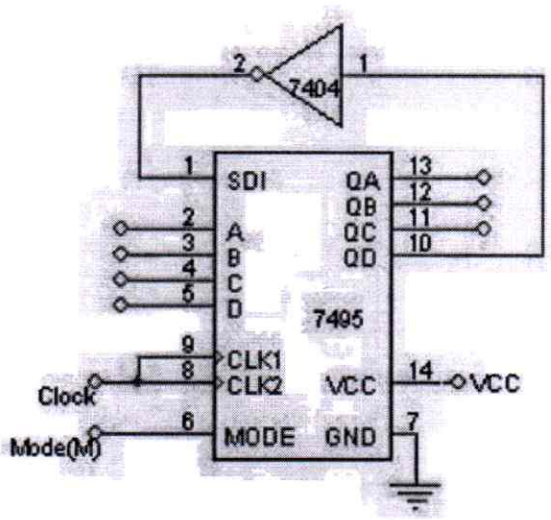
1. Connections are made as per the circuit diagram
2. Apply the data 1000 at A,B,C and D respectively.
3. Keeping the mode M-1, apply one clock pulse.
4. Now the mode M is made 0 and clock pulses are applied one by one and the truth table is verified.
5. Above procedure is repeated for Johnson counter also.

Circuit Diagram: - Ring Counter



Mode	Clock	QA	QB	QC	QD
1	1	1	0	0	0
0	2	0	1	0	0
0	3	0	0	1	0
0	4	0	0	0	1
0	5	1	0	0	0
0	6	repeats			

Johnson Counter:-



Mode	Clock	QA	QB	QC	QD
1	1	1	0	0	0
0	2	1	1	0	0
0	3	1	1	1	0
0	4	1	1	1	1
0	5	0	1	1	1
0	6	0	0	1	1
0	7	0	0	0	1
0	8	0	0	0	0
0	9	1	0	0	0
0	10	repeats			

Result: Thus Designed and tested the Ring Counter/Johnson Counter.

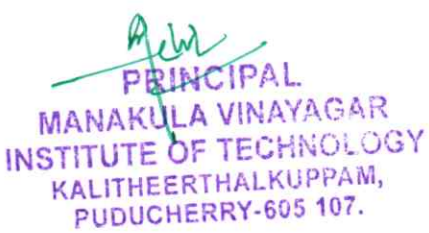
PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

P 101 - COMPUTER PROGRAMMING LAB

List of Exercises

1. Study of OS Commands
2. Write a C program to find the Area of the triangle.
3. Write a C program to find the total and average percentage obtained by a student for 6 subjects.
4. Write a C program to read a three digit number and produce output like
   1 hundreds
   7 tens
   2 units for an input of 172.
5. Write a C program to check whether a given character is vowel or not using Switch – Case statement.
6. Write a C program to print the numbers from 1 to 10 along with their squares.
7. Write a C program to find the sum of 'n' numbers using for, do – while statements.
8. Write a C program to find the factorial of a given number using Functions.
9. Write a C program to swap two numbers using call by value and call by reference.
10. Write a C program to find the smallest and largest element in an array.
11. Write a C program to perform matrix multiplication.
12. Write a C program to demonstrate the usage of Local and Global variables.
13. Write a C program to perform various string handling functions: strlen, strcpy, strcat, strcmp.
14. Write a C program to remove all characters in a string except alphabets.
15. Write a C program to find the sum of an integer array using pointers.
16. Write a C program to find the Maximum element in an integer array using pointers.
17. Write a C program to create student details using Structures.
18. Write a C program to display the contents of the file on the monitor screen.
19. Create a File by getting the input from the keyboard and retrieve the contents of the file using file operation commands.
20. Write a C program to pass the parameter using command line arguments



IT-P31 DATA STRUCTURES LAB

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P31	DATA STRUCTURES LAB	0	0	3

Course Objectives:

1. To introduce the basics of C++ programming language.
2. To introduce the concepts of ADTs.
3. To introduce the concepts of Hashing and Sorting.
4. Solving various problems using techniques introduced in this course
5. Analyze the algorithm's / program's efficiency in terms of time and space complexity

Course Outcomes:

On successful completion of this practical's students will be able to:

1. Solve any given problem by identifying appropriate Data Structure
2. Evaluate program's efficiency in terms of time and space complexity

Exercises:

1. Programs using C++ concepts like
   - classes & objects
   - Constructors & Destructors
   - Function Overloading
   - Inheritance
   - Operator overloading
   - Polymorphism & virtual functions
   - I/O streams
   - File operations.
   - Templates
   - Exception handling (to be included in all problems)
   - String operations

2. Programs related to data structures using C++

Implementation of Sorting techniques
Implementation of Searching techniques
Implementation of stack and queue operations using linked list and array.
Expression evaluation
Polynomial addition
Sparse matrix addition
Binary tree representation and traversal techniques
Binary search trees
Graph representation and traversal techniques
Single source shortest path algorithm
Hashing and collision resolution techniques
AVL Trees

Content beyond Syllabus:

1. Analyze program's efficiency in terms of time and space complexity

Text Books:

3. Mark Allen Weiss, Data structures and algorithm analysis in C++, Pearson Education, 6th edition, 2011
4. YedidyahLangsam, Moshe J Augenstein and Aaron M Tanenbaum, Data Structures using C and C++, 2nd edition, Prentice Hall of India, 2009.

Reference Books:

1. G.A.V.Pai, Data Structures and Algorithms – Concepts, Techniques and Applications, Tata McGraw Hill Publishing Company Limited, New Delhi, 2008.
2. Ellis Horowitz and SartajSahni, Fundamentals of Data structures, Galgotia Publications, 2nd Edition, New Delhi, 2001.
3. Alfred V. Aho, Jeffrey D. Ullman, John E. Hopcroft. Data Structures and Algorithms. Addison Wesley, 1983

Websites:

1. http://www.cs.sunysb.edu/~skiena/214/lectures/
2. http://cse.yeditepe.edu.tr/~odemir/spring2012/cse211/analysis.pdf

MANAKULAVINAYAGARINSTITUTEOFTECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENTOFINFORMATIONTECHNOLOGY

IT–P31 DATA STRUCTURE LABORATORY

Mini Project

on

SIMPLE GRAPH USING THE ADJACENCY LIST& DEPTH FIRST SEARCH TRAVERSAL

Submittedby

DHEEPASRI .G
DINESHKUMAR .S
DIVYADHARSHINI.S A
GOGUL. S.S
GOKUL.S
GOKULAN.S
GUNASELAN S
JOHNFERNANDAS.J
KARTHIKEYAN.K
KAVYA .V
KISHOR.K
KISHORE.M
LOGA JANANI .R

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
Puducherry
605 107

IIYEAR–IT

(2020-2021)

SIMPLEGRAPHUSINGTHEADJACENCYLIST

AIM:

TowriteaC++programtoimplementanddemonstrateasimplegraphusingtheadjacencylist.

DESCRIPTION:

Herewearegoingtodisplaytheadjacencylistforaweighteddirectedgraph.Wehaveusedtwo structures to hold the adjacency list and edges of the graph. The adjacency list is displayed as (start_vertex, end_vertex, weight).

ALGORITHM:

Step1:Starttheprogram.
Step2:Createthestructuretohold theadjacencyandedgesof the graph.
Step3:CreateaclassDiaGraphtoinsertnewnodesintoadjacencylistfromgivengraph. Step 4: Insert the new nodes into the graph.
Step5:Displaythealladjacentverticesofgivenvertex. Step 6:Stop the program

SOURCECODE:

```
#include <iostream>
usingnamespacestd;
//storesadjacencylistitems
structadjNode {
   intval, cost;
   adjNode*next;
};
//structuretostoreedges
structgraphEdge {
   intstart_ver,end_ver,weight;
};
classDiaGraph{
   //insertnewnodesintoadjacencylistfromgivengraph
   adjNode*getAdjListNode(intvalue,intweight,adjNode*head){
      adjNode* newNode = new adjNode;
      newNode->val = value;
      newNode->cost=weight;

      newNode->next=head;//pointnewnodetocurrenthead return
      newNode;
   }
   intN;//numberofnodesin thegraph
```

DEPTHFIRSTSEARCHTRAVERSAL

AIM:

ToimplementtheDFS(DepthFirstSearch)traversaltechniqueusingC++.

DESCRIPTION:

Unlike BFS in which we explore the nodes breadthwise, in DFS we explore the nodes depth-wise.InDFSweuseastackdatastructureforstoringthenodesbeingexplored.Theedgesthat lead us to unexplored nodes are called 'discovery edges' while the edges leading to already visited nodes are called 'block edges'.

ALGORITHM:

Step1: Insertttherootnodeorstartingnodeofatreeoragraphinthestack.
Step2: Popthetopitemfromthestackand addittothevisited list.
Step3: Findalltheadjacentnodesofthenodemarkedvisitedandaddtheonesthatare not yet visited, to the stack.
Step4:Repeatsteps2and3untilthestackis empty.\

SOURCECODE:

```
#include<iostream>
#include <list>
usingnamespacestd;
//graphclassforDFStravesal
class DFSGraph
{
intV;    //No.ofvertices
list<int>*adjList;    //adjacencylist
voidDFS_util(intv,boolvisited[]);//AfunctionusedbyDFS public:
   //classConstructor
DFSGraph(int V)
   {
 this->V= V;
 adjList=new list<int>[V];
   }
   //functiontoaddanedgetograph void
addEdge(int v, int w){
adjList[v].push_back(w);//Addwtov'slist.
   }

void DFS();    //DFStraversalfunction
};
voidDFSGraph::DFS_util(intv,boolvisited[])
{
   //currentnodevisvisited
```

Subject Code	Subject Name	Lectures (Periods)	Tutorial (Periods)	Practical (Periods)
IT-P32	ELECTRONIC DEVICES AND CIRCUITS LAB	0	0	3

Course Objectives:

1. Introduce the basic concepts of various electronic circuits.
2. To study the performance of various types of feedback amplifiers.
3. To analyze and test the performance of small signal and large signal amplifiers.
4. To test and examine the applications of operational amplifiers.

Course Outcomes:

On successful completion of the lab classes students will be able to,

1. Conceptually and fully aware of the basic concepts, techniques and applications of electronic circuits.
2. To enhance their technical skills through analyzing the waveforms obtained at various stages of the circuit.
3. Carry out design of the various electronic circuits suitable for a specific application.

- **Experiment List:**
- VI characteristics of semiconductor diodes.
- Diode clipping and clamping circuits.
- Characteristics of CB transistor configuration.
- Input and Output characteristics of CE transistor configuration.
- Characteristics of FET, Determination of drain resistance, mutual conductance and amplification factor.
- Feedback amplifier, To determine the frequency response with and without feedback.
- Hartley oscillator and Wein-bridge oscillator.
- Class B push-pull power amplifier.
- Applications of OP-Amps - Adder, Subtractor, Integrator and Differentiator.
- Active low pass and high pass filters using OP-AMP.

Text Books:

1. Jacob Millman and C. Halkias, Satya brataJit, Electronic Devices and circuits, Second edition, McGraw Hill Publications, 2007.

2. Jacob Millman and Arvin Grabel, Micro-Electronics, McGraw Hill, Fifth edition, 2008.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
DEPARTMENT OF IT

VOLTAGE DIVIDER BIAS USING MULTISIM

Submitted by

MADHUMIDHA .B
MIRUNALINIDEVI.N
MOHAMED THALIF.M
MOHAN SHYAME
PRAKASHRAJ. M
PRAVEEN KUMAR.E

II YEAR IT

(2020 2021)

VOLTAGE DIVIDER BIAS USING MULTISIM

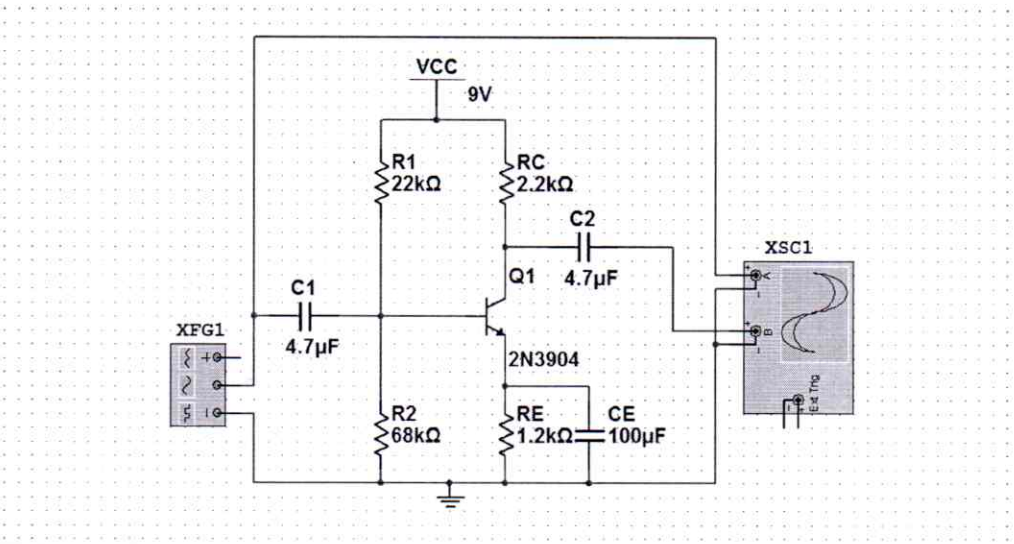
Aim:

Biasing is the process of providing DC voltage which helps in the functioning of the circuit. A transistor is based in order to make the emitter base junction forward biased and collector base junction reverse biased, so that it maintains in active region, to work as an amplifier.

Software Required:

Multi-sim Tools

Schematic circuit:



CIRCUIT OPERATION:

Voltage Divider Bias Circuit, also known as **emitter current bias,** is the most stable of the three basic transistor bias circuits. A voltage divider bias circuit, and the current and voltage conditions throughout the circuit. It is seen that, as well as the collector resistor (R_C), there is an emitter resistor (R_E) connected in series with the transistor. The total dc load in series with the transistor is ($R_C + R_E$), and this total resistance must be used when drawing the dc load line for the circuit. Resistors R_1 and R_2 constitute a voltage divider that divides the supply voltage to produce the base bias voltage (V_B).

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

P103 - BASIC ELECTRICAL AND ELECTRONICS LAB

Electrical Laboratory Experiments

1. Electrical Safety, Precautions, study of tools and accessories.
2. Practices of different joints.
3. Wiring and testing of series and parallel lamp circuits.
4. Staircase wiring.
5. Doctor's room wiring.
6. Bed room wiring.
7. Godown wiring.
8. Wiring and testing a ceiling fan and fluorescent lamp circuit.
9. Study of different types of fuses, circuits breakers and A.C and D.C meters

Electronics Laboratory Experiments

1. Study of CRO

(a) Measurement of AC and DC voltages

(b) Frequency and phase measurements (using Lissajou's figures)

2. Verification of Kirchoff's Voltage and Current Laws

Determine the voltage and current in given circuits using Kirchoff's laws theoretically and verify the laws experimentally.

3. Characteristics and applications of PN junction diode.

Forward and Reverse characteristics of PN junction diode.

Application of Diode as Half wave Rectifier – Measurement of ripple factor with and without capacitor filter

4. Frequency Response of RC Coupled Amplifiers

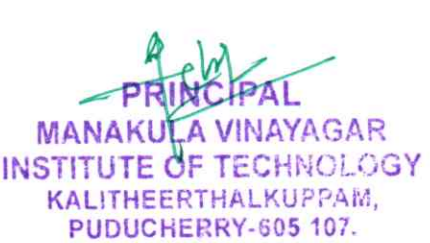
Determination of frequency response of given RC coupled amplifier - Calculation of bandwidth.

5. Study of Logic Gates

(a) Verification of Demorgan's theorems

(b) Verification of truth tables of OR, AND, NOT, NAND, NOR, EX-OR, EX-NOR gates and Flipflops - JK, RS, T and D

(c) Implementation of digital functions using logic gates and Universal gates



IT-P33 DIGITAL LAB

Subject Code	Subject Name	Lectures (Periods)	Tutorial (Periods)	Practical (Periods)
IT-P33	DIGITAL LAB	0	0	3

Course Objectives:

1. To perform fundamental operations on digital circuits.
2. To apply the concepts of basic combinational logic circuits, sequential circuit elements, and programmable logic in the laboratory setting.
3. To design the combinational and sequential circuits using Verilog Hardware Description Language (VHDL)

Course Objectives:

- To apply knowledge of number systems, codes and Boolean algebra to the analysis and design of digital logic circuits.
- To identify, formulate, and solve engineering problems in the area of digital logic circuit design.
- To use the techniques, skills, and modern engineering tools such as logic works and VHDL, necessary for engineering practice.
- To design a digital system, components or process to meet desired needs within realistic constraints

List of Experiments:

1. Implementation of logic circuits using gates
   - Full adder/full subtractor
   - Implementation of logic functions using universal gates only
   - Code converters
   - Parity generator and Checker
   - Design of priority encoder
   - Implementation of Boolean functions using MUX
   - Design of decoder, Demultiplexer.
2. Implementation of circuits using MSI
   - Synchronous counters
   - Asynchronous counters
   - Binary multiplier
   - Decimal Adder
   - Universal shift register
   - Design of Arithmetic unit
3. Interface experiments with MSI
   - Interface of ALU with memory
4. Design and Implementation of combinational circuits using Verilog Hardware Description Language (VHDL)
   - Combinational circuits – Adder/ Subtractor, Binary multiplier
   - Sequential circuits – Flip flops, counters.

Text Books:

1. Morris Mano, Digital Design, Third Edition, Pearson Education, 2002.
2. Carl Hamacher, ZvonkoVranesic And SafwatZaky, Computer Organization, Fifth Edition, Tata McGraw Hill, 2002.

Reference Books:

1. Charles H. Roth, Jr., Fundamentals of Logic Design, Fifth Edition, Jaico Publishing House, 2003.
2. William Stallings, Computer Organization and Architecture – Designing for Performance, Sixth Edition, Pearson Education, 2003.
3. David A. Patterson And John L. Hennessy, Computer Organization and Design: The Hardware/Software Interfacel, Third Edition, Elsevier, 2005.
4. John P. Hayes, Computer Architecture and Organization, Third Edition, Tata McGraw Hill, 1998.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT P33 Digital Lab

Mini Project on Implementation of Parallel Adder

Submitted by

SNEHA.U, SRINIVASSAN

SUBALAKSHMI.S, SUDHARSAN

SUBASHINI.G, SUBHASH.B

SUHAIB AHAMED K.A

SURIYAPRIYA, SUVEDHA.M

VANDHANA .K, YOKESH.R

VASANTHA GANESH. R

YAZHINI SIVASHANKARI .B

II YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

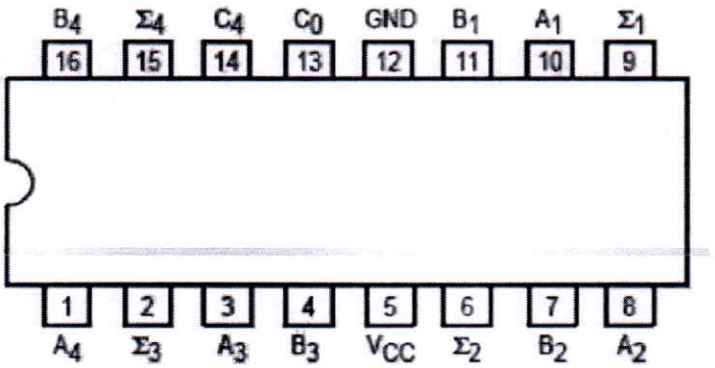
Aim – Implementation of 4-Bit Parallel Adder Using 7483 Ic.

APPRATUS REQUIRED – Digital trainer kit, IC 7483 (4-bit parallel adder).

BRIEF THEOR - A 4-bit adder is a circuit which adds two 4-bits numbers, say, A and B. In addition, a 4-bit adder will have another single-bit input which is added to the two numbers called the carry-in (C_{in}). The output of the 4-bit adder is a 4-bit sum (S) and a carry-out (C_{out}) bit.

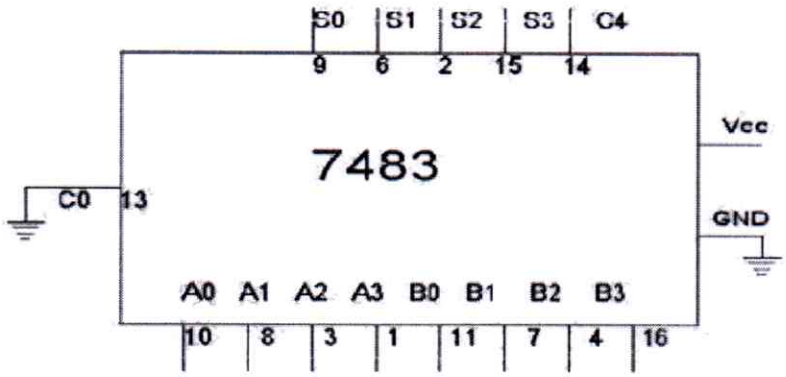
PIN CONFIGURATION–

Pin Diagram of IC 7483



LOGIC DIAGRAM:-

7483 4-bit Parallel Adder



Truth table of 4-bit parallel adder

A3	A2	A1	A0	B3	B2	B1	B0	C4 (V)	S3(V)	S2(V)	S1(V)	S0(V)
0	0	0	1	0	0	1	0	0	0	0	1	1
0	1	0	1	1	0	1	1	1	1	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	0	0	1	1	0	1	0	1	0

PROCEDURE –

a) Make the connections as per the logic diagram.
b) Connect +5v and ground according to pin configuration.
c) Apply diff combinations of inputs to the i/p terminals.
d) Note o/p for summation.
e) Verify the truth table.

RESULT- Binary 4-bit full adder is studied and verified.

IT-T42 COMMUNICATION ENGINEERING-I

Course Objectives:

1. To introduce the basics of electronic communication s
2. To introduce different analog modulation systems.
3. To introduce the operation of modulator and demodulator for different analog modulation systems.
4. To explore the use of pulse modulation system
5. To introduce the techniques of digital modulation.

Course Outcomes:

On successful completion of the course students:

1. Will be clear with the concepts of different analog modulation systems
2. Will understand the need for pulse modulation systems
3. Will have a clear idea on concept and applications of digital modulation systems

Unit I: Amplitude Modulation Systems **(12 Periods)**

External and internal noise - Noise figure - Need for modulation - Amplitude modulation –Frequency spectrum of AM wave – Representation of AM – Power relation – Generation of AM waves – Balanced modulators – Suppression of carrier – DSB and SSB –Demodulation of AM waves – Synchronous and envelope detectors.

Unit II: Angle Modulation System **(10 Periods)**

Frequency modulation and phase modulation – Mathematical representation of FM – Frequency spectrum of FM wave – Generation of FM wave – Direct and Indirect methods – Demodulation of FM waves – Slope detector –Balanced slope detector – Foster-Seeley discriminator – Ratio detector.

Unit III: Transmitters and Receivers **(10 Periods)**

Low level and high level AM transmitters – FM transmitter – Super heterodyne AM receiver – Receiver characteristics - Communication receiver – Diversity reception – FM receivers.

Unit IV: Pulse Modulation **(9 Periods)**

Principles of pulse modulation – sampling theorem, PAM – PWM – PPM— Generation of PAM, PPM and PWM waves – Demodulation of PAM, PWM and PPM. Principle of Pulse code modulation - elements of PCM system- Delta modulation and DPCM-transmitter and receiver

Unit V: Digital Communication **(9 Periods)**

Principle of ASK- Transmitter and receiver for coherent BPSK, BFSK and QPSK- Principle of QAM - transmitter and receiver for 8- QAM and 8-PSK. Basic principle of M-ary PSK and M-ary FSK. Bandwidth efficiency and error performance comparisons of PSK, FSK and QAM (detailed derivations not required)

Content beyond the Syllabus:

Students will be motivated to visit the websites of AIR and Doordharshan and understand the practical frequency assignment, broadcast power level and coverage area of all the Indian radio and TV stations

Text Books:

1. George Kennedy and Bernard Davis, Electronic Communication Systems, Fourth edition, Tata Mc Graw Hill, 2008.
2. Simon Haykin, Communication Systems, Fourth edition, Wiley, 2013

Reference Books:

1. Wayne Tomasi, Electronic Communication Systems, Fifth edition, Pearson Education, 2008.
2. D. Roddy and Coolen, Electronic Communications, Fourth edition, Pearson Education, 2008.

Website:

1. http://drdo.gov.in/drdo/labs/LRDE/English

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF IT

Mini project on Performance analysis of digital modulation schemes in AWGN channnel

Submitted by

ABINESH.S
AISWARIYA.S
ANITHA.K
ARAVINDH.M
BHARATHI.P
DEVI.P

II YEAR IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY 605 107

Performance analysis of digital modulation schemes in AWGN channel

AIM:

To write MATLAB programs to simulate the digital modulation schemes BPSK, QPSK and 8 - PSK and study their BER performance in AWGN channels.

PROGRAM:

```
clc;
clear all;
close all;
M=2;
N=10^5;
data=randint(1,N,M);
mod_op=pskmod(data,M);
ber=[];
for snr=0:0.5:20
ch_op=awgn(mod_op,snr);
demod=pskdemod(ch_op,M);
b=symerr(data,demod);
ber=[ber b];
end
snr=0:0.5:20;
semilogy(snr,ber,'r*-','LineWidth',2);
holdon;
M=4;
N=10^5;
data=randint(1,N,M);
mod_op=pskmod(data,M);
ber=[];
for snr=0:0.5:20
ch_op=awgn(mod_op,snr);
demod=pskdemod(ch_op,M);
b=symerr(data,demod);
ber=[ber b];
end
snr=0:0.5:20;
semilogy(snr,ber,'mo-','LineWidth',2);
holdon;
M=8;
N=10^5;
```

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM,
PUDUCHERRY 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

IT-T43 DESIGNANDANALYSISOFALGORITHMS

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T43	Design and Analysis of Algorithms	3	1	0

Course Objectives:

1. To introduce the fundamental strategies of different algorithm design techniques.
2. Solving various problems using techniques introduced in this course.
3. Analyze the algorithm's / program's efficiency in terms of time and space complexity.

Course Outcomes:

On successful completion of this course students will be able to:

1. Analyze / compare the given algorithm.
2. Compute the time complexity/space complexity of any recursive/non recursive algorithms.
3. Solve any given problem using the fundamental design techniques.

Unit: I **(12 Periods)**

Introduction: what is an Algorithm – contradiction- mathematical induction -Efficiency of algorithms – average and worst-case – the order of - asymptotic notation.

Analysis Of Algorithms: Analyzing control structures – solving recurrences – homogeneous recurrences – inhomogeneous recurrences.

Unit: II **(12 Periods)**

Divide And Conquer Method: General method - Binary search – finding maximum and minimum - merge sort – quick sort – strassen's matrix multiplication.

Greedy Method: General method - Knapsack problem – job sequencing with deadlines - Prim's algorithm – Kruskal's algorithm – optimal storage on tapes – optimal merge patterns - Dijkstra's algorithm.

Unit: III **(12 Periods)**

Dynamic Programming: General method –Principle of optimality – multi stage graph - all pairs shortest paths - Warshall's and Floyd's algorithms – optimal binary search tree – 0 / 1 knapsack problem – traveling salesman problem

Unit: IV **(12 Periods)**

Tree traversals: Depth first search – articulation points – breadth first search

Backtracking: General method - n queen's problem – sum of subsets – graph coloring – Hamiltonian cycle – knapsack problem

Unit: V **(12 Periods)**

Branch And Bound: Least Cost search – 15 puzzle – control abstractions for LC search – bounding – FIFO Branch and bound – LC branch and Bound - Knapsack problem: LC branch and bound – FIFO branch and bound solutions – Traveling salesman problem – assignment problem

(Total : 60 Periods)

Content beyond Syllabus:

1. Algebraic problems
2. NP Hard and NP complete problems
3. Approximation Algorithms

Text Books:

1. Gilles Brassard and Paul Brately, Fundamentals of Algorithmics, Prentice Hall of India, 1997.
2. AnanyLevitin, Introduction to Design and Analysis of Algorithms, Pearson Education Inc., 2005.
3. Ellis Horowitz, SartajSahni and S. Rajasekaran, Fundamentals of Computer Algorithms , Galgotia Publications, 2nd Edition, New Delhi, 2003.

Reference Books:

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – T43 DESIGN AND ANALYSIS OF ALGORITHMS

Mini Project on

HAMILTONIAN CYCLE & TOWER OF HANOI

Submitted by

KALYANI VARSHINI. B
KEERTHANA.S
LAVANYA.S
LOGANATHAN .V
MAHADEVAN . R

II YEAR – IT

(2020-2021)

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHAI KUPPAM,
PUDUCHERRY 605 107

HAMILTONIAN CYCLE

AIM:

To write a c++ program to implement Hamiltonian cycle.

DESCRIPTION:

A Hamiltonian cycle is a Hamiltonian Path such that there is an edge (in graph) from the last vertex to the first vertex of the Hamiltonian Path. It is in an undirected graph is a path that visits each vertex of the graph exactly once

ALGORITHM:

Begin

1. function isSafe() is used to check for whether it is adjacent to the previously added vertex and already not added.
2. function hamiltonianCycle() solves the hamiltonian problem.
3. function hamCycle() uses hamiltonianCycle() to solve the hamiltonian problem. It returns false if there is no Hamiltonian Cycle possible, otherwise return true and prints the path.

End

PROGRAM:

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#define N 5
using namespace std;
void displaytheSolution(int path[]);
bool isSafe(int n, bool g[N][N], int path[], int pos) {
  if (g [path[pos-1]][n] == 0)
    return false;
```

TOWER OF HANOI

AIM:

To write a c++ program to implement Tower of Hanoi

DESCRIPTION:

Tower of Hanoi, is a mathematical puzzle which consists of three towers (pegs) and more than one rings . These rings are of different sizes and stacked upon in an ascending order, i.e. the smaller one sits over the larger one. There are other variations of the puzzle where the number of disks increase, but the tower count remains the same.Tower of Hanoi puzzle with n disks can be solved in minimum 2^n-1 steps.

ALGORITHM:

START
Procedure Hanoi(disk, source, dest, aux)
IF disk == 1, THEN
move disk from source to dest
ELSE
Hanoi(disk - 1, source, aux, dest) // Step 1
move disk from source to dest // Step 2
Hanoi(disk - 1, aux, dest, source) // Step 3
END IF
END Procedure
STOP

PROGRAM:

```
#include<iostream.h>
#include<conio.h>
using namespace std;
void TOH(int d, char t1, char t2, char t3)
{
if(d==1)
{
cout<<"\nShift top disk from tower"<<t1<<" to tower"<<t2;
return;
}
TOH(d-1,t1,t3,t2);
cout<<"\nShift top disk from tower"<<t1<<" to tower"<<t2;
TOH(d-1,t3,t2,t1);
}
int main()
```

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – T43 DESIGN AND ANALYSIS OF ALGORITHMS

Mini Project on

STRASSEN'S MATRIX MULTIPLICATION & FLOYD WARSHALL ALGORITHM

Submitted by

GAJALAKSHMI R
GANESH .T
GURUPRASAD. G
HEMAVATHI J
JAYA PRIYA.S
KALAIVANI. M

II YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY 605 107

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

STRASSEN'S MATRIX MULTIPLICATION

AIM:

To write a C++ program to implement Strassen matrix multiplication.

ALGORITHM:

Matrix-Multiplication (X, Y, Z)
for i = 1 to p do
  for j = 1 to r do
    Z[i,j] := 0
    for k = 1 to q do
      Z[i,j] := Z[i,j] + X[i,k] × Y[k,j]

DESCRIPTION:

This C ++program implements Strassen's algorithm to multiply two matrices. This is a program to compute product of two matrices using Strassen Multiplication algorithm. Here the dimensions of matrices must be a power of 2.
Here is the source code of the C++program to multiply 2*2 matrices using Strassen's algorithm. The C program is successfully compiled.

SOURCE CODE:

```
#include <bits/stdc++.h>
using namespace std;
typedef long long lld;

/* Strassen's Algorithm for matrix multiplication
  Complexity:    O(n^2.808) */

inline lld** MatrixMultiply(lld** a, lld** b, int n,
                            int l, int m)
{
   lld** c = new lld*[n];
   for (int i = 0; i < n; i++)
      c[i] = new lld[m];

   for (int i = 0; i < n; i++) {
      for (int j = 0; j < m; j++) {
         c[i][j] = 0;
         for (int k = 0; k < l; k++) {
            c[i][j] += a[i][k] * b[k][j];
         }
      }
   }
   return c;
}
```

FLOYD WARSHALL ALGORITHM

AIM:

To write a c++ program to implement Floyd warshall algorithm.

ALGORITHM:

let dist be a |V| × |V| array of minimum distances initialized to ∞ (infinity)

```
for each vertex v
  dist[v][v] ← 0
for each edge (u,v)
  dist[u][v] ← w(u,v)  // the weight of the edge (u,v)
for k from 1 to |V|
  for i from 1 to |V|
    for j from 1 to |V|
      if dist[i][j] > dist[i][k] + dist[k][j]
        dist[i][j] ← dist[i][k] + dist[k][j]
      end if
```

DESCRIPTION:

Floyd-Warshall Algorithm is an algorithm for finding the shortest path between all the pairs of vertices in a weighted graph. This algorithm works for both the directed and undirected weighted graphs. But, it does not work for the graphs with negative cycles (where the sum of the edges in a cycle is negative).

IT-T44 MICROPROCESSORS AND MICROCONTROLLERS

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T44	Microprocessors and Microcontrollers	3	1	0

Pre-requisite:
Digital Electronics and Computer Architecture

Course Objectives:

1. To understand the architectures and the instruction set of 8085 microprocessor
2. To understand the architectures and the instruction set of 8086 microprocessor
3. To understand the architectures and the instruction set of 8051 microcontroller
4. To learn the assembly language program using 8085, 8086 and 8051 instruction set
5. To learn interfacing of microprocessors and microcontrollers with various peripheral

Course Outcomes:

On successful completion of this course students will be able to:

1. Understanding the inner working components of the microprocessor and microcontrollers
2. Developing assembly language program using 8085 instruction set
3. Developing assembly language program using 8086 instruction set
4. Developing assembly language program using 8051 instruction set
5. Developing various I/O programs for 9085, 8086 and 8051

Unit: I

Intel 8085 Microprocessor: Introduction - Need for Microprocessors – Evolution – Intel 8085 Hardware - Architecture – Pin description - Internal Registers – Arithmetic and Logic Unit – Control Unit – Instruction word size - Addressing modes – Instruction Set – Assembly Language Programming - Stacks and Subroutines - Timing Diagrams. Evolution of Microprocessors – 16-bit and 32-bit microprocessors.

Unit II

Intel 8085 Interrupts and DMA: 8085 Interrupts – Software and Hardware Interrupts – 8259 Programmable Interrupt Controller - Data Transfer Techniques – Synchronous, Asynchronous and Direct Memory Access (DMA) and 8237 DMA Controller- 8253 Programmable Interval Timer.

Unit III

Memory & I/O Interfacing: Types of memory – Memory mapping and addressing – Concept of I/O map – types – I/O decode logic – Interfacing key switches and LEDs – 8279 Keyboard/Display Interface - 8255 Programmable Peripheral Interface – Concept of Serial Communication – 8251 USART – RS232C Interface.

Unit IV

Intel 8086 Microprocessor: Introduction-Intel 8086 Hardware – Pin description – External memory Addressing – Bus cycles – Interrupt Processing. Addressing modes - Instruction set – Assembler Directives.

Unit V

Microcontroller: Intel 8051 Microcontroller: Introduction – Architecture – Memory Organization – Special Function Registers – Pins and Signals – Timing and control – Port Operation – Memory and I/O interfacing – Interrupts – Instruction Set and Programming.

(Total : 60 Periods)

Content beyond Syllabus:

Study of Multi-core Architecture and Programming
Study of Intel i5 processor

Text Books:

1. Ramesh S. Gaonkar, "Microprocessor Architecture, Programming and Applications with 8085", Penram International Publications, Fifth Edition,2002.
2. Krishna Kant, "Microprocessors and Microcontrollers – Architectures, Programming and System Design 8085, 8086, 8051, 8096", PHI, 2008.
3. N. Senthil Kumar, M Saravanan and S. Jeevananthan, "Microprocessors and Microcontrollers", Oxford University Press, 2010.

Reference Books:

1. A. P. Godse and D.A Godse, "Microprocessors and Microcontrollers", Technical Publications, Fourth Edition, 2008.
2. Barry B. Brey, "The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386 and 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III Pentium 4 – Architecture, Programming and Interfacing, 7th Edn., PHI, 2008.
3. Ajay V Deshmukh, "Microcontrollers – Theory and Applications", Tata McGraw-Hill, Seventh Edition, 2007.

Websites:

1. http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0211a/index.html
2. http://www.arm.com/products/processors/classic/arm7/index.php
3. http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0211a/index.html
4. **http://www.embeddedindia.com/**

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – T44 MICROPROCESSORS AND MICROCONTROLLERS

Mini Project on Fading a LED

Submitted by

MALINI.P

MOHAMED AKHIL.R

MOHAMED FAGAD.I

MOHAMED THALHAA.F

MOUNISHAA.P

NANTHINI.N

NIVAAS.D

PATHRINATH M

PRAVEEN.V

SATHIASEELAN B S

SHANMUGA PRIYA. S

II YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHAI KUPPAM,
PUDUCHERRY - 605 107

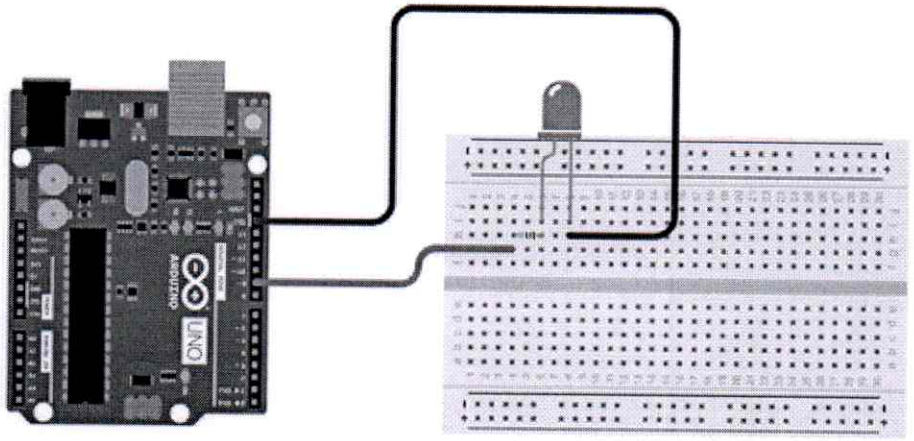
Demonstrates the use of the analogWrite() function in fading an LED off and on. AnalogWrite uses pulse width modulation (PWM), turning a digital pin on and off very quickly with different ratio between on and off, to create a fading effect.

Hardware Required

- Arduino board
- LED
- 220 ohm resistor
- hook-up wires
- breadboard

CIRCUIT

Connect the **anode** (the longer, positive leg) of your LED to digital output pin 9 on your board through a 220 ohm resistor. Connect the **cathode** (the shorter, negative leg) directly to ground.



IT-T45 JAVA PROGRAMMING

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T45	Java Programming	3	1	---

Pre-requisite: Object Oriented Programming

Course Objectives:

1. To understand the basics of Java
2. To learn the features of Java
3. To learn the advanced concepts in Java.

Course Outcomes: Students will understand the benefits and capabilities of Java.

Unit: I

Creation of Java, importance of Java to internet, byte code, Java buzzwords, data types, declaring variables, dynamic initialization, scope and life time of variables, arrays, operators, control statements, type conversion and casting, compiling and running of simple Java program. Concepts of classes and objects, class fundamentals Declaring objects, assigning object reference variables, introducing methods, constructors, usage of static with data and methods, usage of final with data, access control, this key word, garbage collection, overloading methods and constructors, parameter passing - call by value, recursion, nested classes and inner classes, exploring the String class.

Unit: II

Basic concepts, member access rules, usage of super key word, forms of inheritance, method overriding, abstract classes, dynamic method dispatch, using final with inheritance, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.

Unit: III

Concepts of Exception handling, types of exceptions, usage of try, catch, throw, throws and finally keywords, Built-in exceptions, creating own exception sub classes, Concepts of Multithreading, differences between process and thread, thread life cycle ,creating multiple threads using Thread class, Runnable interface, Synchronization, thread priorities, inter thread communication, daemon threads, deadlocks, thread groups.

Unit: IV

Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes.

AWT : Concepts of components, container, panel, window, frame, canvas, Font class, Color class and Graphics.

Applets - Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.

Unit: V

RMI- JDBC- Developing Java Program for RMI and JDBC.

(Total : 60 Periods)

Content beyond Syllabus:

Java's support in graphics, networking, web technology etc.

Text Books:

1. The Complete Reference Java J2SE 5th Edition, Herbert Schildt, TMH Publishing Company Ltd, NewDelhi.
2. Big Java 2nd Edition, Cay Horstmann, John Wiley and Sons.

Reference Books:

1. Java How to Program, Sixth Edition, H.M.Dietel and P.J.Dietel, Pearson Education/PHI
2. Core Java 2, Vol 1, Fundamentals, Cay.S.Horstmann and Gary Cornell, Seventh Edition, Pearson Education.
3. Core Java 2, Vol 2, Advanced Features, Cay.S.Horstmann and Gary Cornell, Seventh Edition, Pearson Education.

Websites:

1. http://www.ibm.com/developerworks/java/
2. http://docs.oracle.com/javase/tutorial/rmi/.
3. IBM's tutorials on Swings, AWT controls and JDBC.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-T45 JAVA PROGRAMMING

Java program using Synchronized Threads, demonstrates Producer Consumer concept

Submitted by

VERONICA T

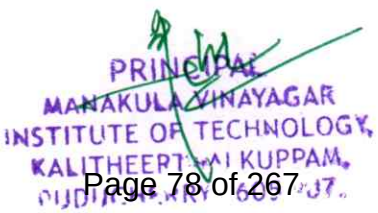
VIDYA LAKSHMI. B

YUVARAJ.S

KAMESH.M

II YEAR – IT

(2020-2021)



Java program using Synchronized Threads that demonstrates Producer Consumer concept.

Producer Consumer problem:

The producer-consumer problem is the classical concurrency of a multi process synchronization problem. It is also known as bound-buffer problem.

The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer used as a queue. The producer generates a piece of data, put it into the buffer and starts again.

ProducerConsumer.java

```
public class ProducerConsumer
{
    public static void main(String[] args)
    {
        Shop c = new Shop();
        Producer p1 = new Producer(c, 1);
        Consumer c1 = new Consumer(c, 1);
        p1.start();
        c1.start();
    }
}
class Shop
{
    private int materials;
    private boolean available = false;
    public synchronized int get()
    {
        while (available == false)
        {
            try
            {
                wait();
            }
            catch (InterruptedException ie)
            {
```

```
            }
        }
        available = false;
        notifyAll();
        return materials;
    }
    public synchronized void put(int value)
    {
        while (available == true)
        {
            try
            {
                wait();
            }
            catch (InterruptedException ie)
            {
                ie.printStackTrace();
            }
        }
        materials = value;
        available = true;
        notifyAll();
    }
}
class Consumer extends Thread
{
    private Shop Shop;
    private int number;
    public Consumer(Shop c, int number)
    {
        Shop = c;
        this.number = number;
    }
    public void run()
    {
        int value = 0;
```

```
        for (int i = 0; i < 10; i++)
        {
            value = Shop.get();
            System.out.println("Consumed value " + this.number+ " got: " + value);
        }
    }
}
class Producer extends Thread
{
    private Shop Shop;
    private int number;

    public Producer(Shop c, int number)
    {
        Shop = c;
        this.number = number;
    }
    public void run()
    {
        for (int i = 0; i < 10; i++)
        {
            Shop.put(i);
            System.out.println("Produced value " + this.number+ " put: " + i);
            try
            {
                sleep((int)(Math.random() * 100));
            }
            catch (InterruptedException ie)
            {
                ie.printStackTrace();
            }
        }
    }
}
```

IT-T46 SYSTEM SOFTWARE

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T46	System Software	3	1	0

Pre-requisite:
Knowledge in C Programming, Assembly languages, Discrete mathematics and data structures

Course Objectives:
1. Understand the design and implementation of Assemblers, loaders, linkers and compilers
2. Understand how source language programs are implemented at the machine level
3. Understand compilation as an instance of language translation.

Course Outcomes:
On successful completion of this course students will be able to:
1. To use of formal attributed grammars for specifying the syntax and semantics of programming languages.
2. Have in depth Working knowledge of the major phases of Loading linking and compiling
3. To design and implement a significant portion of a compiler for a language chosen by the instructor.

Unit: I **(8 periods)**

Introduction to System Software and Machine architecture – Simplified Instructional Computer (SIC)- Traditional machines-VAX Architecture-Pentium Pro Architecture- RISC machines.

Unit: II **(10 periods)**

Assemblers: Basic assembler functions- machine – dependent and machine independent assembler features - Assembler design – Two-pass assembler with overlay structure- one – pass assembler and multi - pass assembler.
Interpreters: Virtual Machine concept- Java Byte Codes- Microsoft Intermediate Language

Unit: III **(9 periods)**

Loaders and Linkers: Basic loader functions, machine – dependent and machine – independent loader features. Loader design – Linkage editors, dynamic linking and bootstrap loaders.

Unit: IV **(9 periods)**

Macro Processors: Functions – Machine independent macro processor features – macro processor design option- Implementation examples.

Unit: V **(9 periods)**

Text editors - Overview of the Editing Process - User Interface – Editor Structure - Interactive debugging systems - Debugging functions and capabilities – Relationship with other parts of the system – User-Interface Criteria. - Introduction to Compilers -Analysis of the source program - The phases- of Compiler.

(Total: 45 Periods)

Content beyond Syllabus:
1. Implementation examples on Assemblers – MASM, SPARC and AIX.
2. Implementation examples on Linkers- MSDOS, SunOS and CRAY MPP.
3. Implementation examples on Compilers- SunOS, GNU NYA Ada Translator, Java compiler and YACC Compiler-. compiler.

Text Books:

1. Leland L Beck and D. Manjula, "System Software",III Edition, Pearson Education , First Impression, 2007.
2. John J Donovan, Systems Programming, Tata McGraw Hill Company, New Delhi, 2004
3. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques, & Tools, 2nd edition Addison-Wesley, 2006.

Reference Books:

1. Dhamdhere D M, Systems Programming and Operating Systems, Tata McGraw Hill Company, New Delhi, 2002.
2. David Galles, Modern Compiler Design, Addison Wesley, 2004.

Websites:
1. http://www.edunotes.in/system-software-notes
2. http://www.uotechnology.edu.iq/sweit/Lectures/Dr-Shaima-Sys-Prog/lec1-2-3-4.pdf

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT T46 System Software

Implementation of Token Separation using C Program

Submitted by

AAKASH.G

AASHIFA.S

AL BISEL.A

ANNUNCIAMARY

ANUSUYA.R

ARUN.D

DHANUJA SRI .P

DHEEPASRI .G

DINESHKUMAR .S

DIVYADHARSHINI.S A

GOGUL. S.S

II YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

AIM:

To write a program to implement the token separation operation

AlGORITHM:

Step 1: Start the program.

Step 2: Store the possible keywords in an array key[][] and their corresponding byte value
in b[ ].

Step 3: Declare all the variables.

Step 4: Declare the file pointer fp for file operation.

Step 5: Open a file sym.c in write mode.

Step 6: Enter valid data into sym.c file until "#" symbol encountered. Then close the file.

Step 7: Open d.c file in read mode.Read the character one by one.

Step 8: If not End of file using switch case check for special symbols.Print the special
symbol.

Step 9: Check whether the string is alphabet or alphanumeric using isalpha() and isalnum() functions.

Step 10: If the string is alphabet assign it to variable "a" and compare with keywords in
array using strcmp() function.

Step 11: If string is keyword print the keyword and its corresponding byte value and copy
the string to variable "data" using strcpy() .

Step 12: Else copy to variable "sym".

Step 13: Check for the character is constant value using isdigti() function and copy the
constant in the variable "val" using strcpy().

Step 14: Print all the datatype, identifier and constant value.

Step 15: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
```

```
#include<string.h>
char key[5][10]={"int","float","char","double"};
int b[5]={2,1,4,8};
int main()
{
int byte;
int label;
int i,j,n,k=0;
char data[10],sym[10],val[10];
char a[20];
char str;
FILE *fp;
clrscr();
fp=fopen("sym.c","w");
printf("\n enter a valid declarations:");
while((str=getchar())!='#')
{
fputc(str,fp);
}
fclose(fp);
fp=fopen("d.c","r");
printf("\n_____");
printf("\t\t SYMBOL TABLE\n");
printf("\n_____");
printf("\ndata-type\tidentifier\tvalue\tbytes-occupied\n");
while((str=fgetc(fp))!=EOF)
{
i=0;
label=0;
switch(str)
{
case ';':
printf("\n\t%d\t%c\t a special symbol",n++,str);
break;
default:
```

```
if(isalpha(str))
{
do{
a[i]=str;
i++;
str=fgetc(fp);
}while(isalpha(str)||isalnum(str));
a[i]='\0';
fseek(fp,-1,1);
for(i=0;i<5;i++)
{
if(strcmp(a,key[i])==0)
{
// printf("\n\t%d\t%s\t a keyword",n++,a);
byte=b[i];
strcpy(data,a);
label=1;
goto aa;
}
}
if(label==0)
{
strcpy(sym,a);
}
}
else if(str=='=')
{
str=fgetc(fp);
if(str=='\")
str=fgetc(fp);
goto aa;
}
else
aa:
if(isdigit(str)||isalpha(str))
```

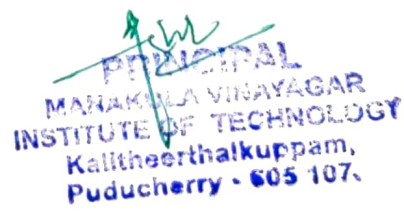
```
{
do{
a[i]=str;
i++;
str=fgetc(fp);
}while(isdigit(str)||str=='.'||isalpha(str));
a[i]='\0';
fseek(fp,-1,1);
strcpy(val,a);
}
}
}
fclose(fp);
getch();
}
```

OUTPUT:

```
enter a valid data:
void main()
{
int a=5;
}
#
```

TOKEN SEPARATION

token no. token name token-type
1 void a keyword
2 main a keyword
3 (a special symbol
4) a special symbol
5 { a special symbol
6 int a keyword
7 a an identifier
8 = an operator
9 5 a constant
10 ; a special symbol
11 } a special symbol



IT-P41 ALGORITHMS LAB

Subject Code	Subject Name	Lectures Periods)	Tutorials (Periods)	Practical (Periods)
IT-P41	Algorithms Lab	0	0	3

Course Objectives:

To introduce the implementation of various design techniques using C and C++.

Course Outcomes:

On successful completion of this course, the students will be able to:

Learn to implement the complex tasks using various design techniques.

Programs to implement the following :

1. Quick sort using divide and conquer
2. Merge sort using divide and conquer
3. Prim's algorithm
4. Kruskal's algorithm
5. Dijikstra's algorithm
6. Optimal binary search tree
7. TSP using dynamic programming.
8. N-queens problem using backtracking.
9. Sum of subsets using backtracking.
10. Graph coloring using backtracking.
11. Hamilton Cycle using backtracking.
12. Knapsack using branch and bound.

Content beyond Syllabus:

1. Advanced data structures and their implementation
2. Implementation of the data structures in different language platforms

Text Books:

1. Robert Sedgewick, Algorithms in C , 3rd Edition, PHI, 2007.

Reference Books:

1. Brian W Kernighan and Dennis M. Ritche, C Programming Language, PHI, 2005.

Websites:

1. http://www.cse.iitd.ernet.in/~ssen/csl356/root.pdf

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-P41 - ALGORITHM LABORATORY

Mini Project On

LONGEST COMMON SUBSEQUENCE & ACTIVITY SELECTION PROBLEM

Submitted by

KISHORE.M
LOGA JANANI .R
MADHUMIDHA .B
MIRUNALINIDEVI.N
MOHAMED THALIF.M

II YEAR – IT

(2020-2021)

LONGEST COMMON SUBSEQUENCE

AIM:

To implement Longest Common Subsequence using c program.

DESCRIPTION:

Here longest means that the subsequence should be the biggest one. The common means that some of the characters are common between the two strings. The subsequence means that some of the characters are taken from the string that is written in increasing order to form a subsequence.

ALGORITHM:

Suppose X and Y are the two given sequences

Initialize a table of LCS having a dimension of X.length * Y.length

XX.label = X

YY.label = Y

LCS[0][] = 0

LCS[][0] = 0

Loop starts from the LCS[1][1]

Now we will compare X[i] and Y[j]

if X[i] is equal to Y[j] then

LCS[i][j] = 1 + LCS[i-1][j-1]

Point an arrow LCS[i][j]

Else

LCS[i][j] = max(LCS[i-1][j], LCS[i][j-1])

SOURCE CODE:

```
#include <stdio.h>
#include <string.h>
int i, j, m, n, LCS_table[20][20];
char S1[20] = "abaaba", S2[20] = "babbab", b[20][20];
void lcsAlgo() {
```

ACTIVITY SELECTION PROBLEM

AIM:

To implement activity selection problem using c program .

DESCRIPTION:

The Activity Selection Problem is an optimization problem which deals with the selection of non-conflicting activities that needs to be executed by a single person or machine in a given time frame.

Each activity is marked by a start and finish time. Greedy technique is used for finding the solution since this is an optimization problem.

ALGORITHM:

Following are the steps we will be following to solve the activity selection problem,

Step 1: Sort the given activities in ascending order according to their finishing time.

Step 2: Select the first activity from sorted array `act[]` and add it to `sol[]` array.

Step 3: Repeat steps 4 and 5 for the remaining activities in `act[]`.

Step 4: If the start time of the currently selected activity is greater than or equal to the finish time of previously selected activity, then add it to the `sol[]` array.

Step 5: Select the next activity in `act[]` array.

Step 6: Print the `sol[]` array.

SOURCE CODE:

```
#include<stdio.h>
int main(){
  int start[] = {1 , 5 , 12};
  int finish[] = {10, 13, 23};
  int activities = sizeof(start)/sizeof(start[0]);
  int i, j;
```

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-P41 - ALGORITHM LABORATORY

Mini Project On

RABIN-KARP ALGORITHM & VERTEX COVER PROBLEM

Submitted by

GOKULAN.S
GUNASELAN S
JOHNFERNANDAS.J
KARTHIKEYAN.K
KAVYA .V
KISHOR.K

II YEAR – IT
(2020-2021)

RABIN-KARP ALGORITHM

AIM : To implement Rabin karp algorithm using c++

DESCRIPTION:

The Naive String Matching algorithm slides the pattern one by one. After each slide, it one by one checks characters at the current shift and if all characters match then prints the match. Like the Naive Algorithm, Rabin-Karp algorithm also slides the pattern one by one. But unlike the Naive algorithm, Rabin Karp algorithm matches the hash value of the pattern with the hash value of current substring of text, and if the hash values match then only it starts matching individual characters.
So Rabin Karp algorithm needs to calculate hash values for following strings:

1) Pattern itself.
2) All the substrings of text of length m.

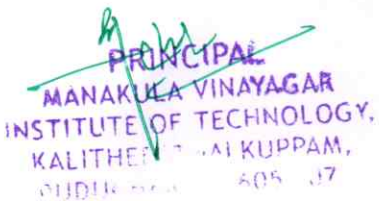
Since we need to efficiently calculate hash values for all the substrings of size m of text, we must have a hash function which has following property.
Hash at the next shift must be efficiently computable from the current hash value and next character in text or we can say hash(txt[s+1 .. s+m]) must be efficiently computable from hash(txt[s .. s+m-1]) and txt[s+m] i.e., hash(txt[s+1 .. s+m])= rehash(txt[s+m], hash(txt[s .. s+m-1])) and rehash must be O(1) operation.
The hash function suggested by Rabin and Karp calculates an integer value. The integer value for a string is numeric value of a string. For example, if all possible characters are from 1 to 10, the numeric value of "122" will be 122. The number of possible characters is higher than 10 (256 in general) and pattern length can be large. So the numeric values cannot be practically stored as an integer. Therefore, the numeric value is calculated using modular arithmetic to make sure that the hash values can be stored in an integer variable (can fit in memory words). To do rehashing, we need to take off the most significant digit and add the new least significant digit for in hash value. Rehashing is done using the following formula.

hash(txt[s+1 .. s+m]) = (d (hash(txt[s .. s+m-1]) - txt[s]*h) + txt[s + m]) mod q
hash(txt[s .. s+m-1]) : Hash value at shift s.
hash(txt[s+1 .. s+m]) : Hash value at next shift (or shift s+1)
d: Number of characters in the alphabet
q: A prime number
h: d^(m-1)

ALGORITHM:

RABIN-KARP-MATCHER (T, P, d, q)

n <- length [T]



VERTEX COVER PROBLEM

AIM : To implement vertex cover problem using c++

DESCRIPTION:

A vertex cover of an undirected graph is a subset of its vertices such that for every edge (u, v) of the graph, either 'u' or 'v' is in the vertex cover. Although the name is Vertex Cover, the set covers all edges of the given graph. Vertex Cover Problem is a known NP Complete problem, i.e., there is no polynomial-time solution for this unless P = NP. There are approximate polynomial-time algorithms to solve the problem though.

ALGORITHM:

1) Initialize the result as {}

2) Consider a set of all edges in given graph. Let the set be E.

3) Do following while E is not empty

...a) Pick an arbitrary edge (u, v) from set E and add 'u' and 'v' to result

...b) Remove all edges from E which are either incident on u or v.

4) Return result

PROGRAM:

```
#include<iostream>
#include <list>
using namespace std;
// This class represents a undirected graph using adjacency list
class Graph
{
    int V;    // No. of vertices
    list<int> *adj;  // Pointer to an array containing adjacency lists
public:
    Graph(int V);  // Constructor
    void addEdge(int v, int w); // function to add an edge to graph
    void printVertexCover();  // prints vertex cover
```

IT-P42 MICROPROCESSORS AND MICROCONTROLLERS LABORATORY

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P42	Microprocessors and Microcontrollers Laboratory	0	0	3

Pre-requisite:
Digital Electronics and Computer Architecture

Course Objectives:

1. To understand the architectures and the instruction set of 8085 microprocessor
2. To understand the architectures and the instruction set of 8086 microprocessor
3. To understand the architectures and the instruction set of 8051 microcontroller
4. To learn the assembly language program using 8085, 8086 and 8051 instruction set
5. To learn interfacing of microprocessors and microcontrollers with various peripheral

Course Outcomes:
On successful completion of this course students will be able to:

1. Understanding the inner working components of the microprocessor and microcontrollers
2. Developing assembly language program using 8085 instruction set
3. Developing assembly language program using 8086 instruction set
4. Developing assembly language program using 8051 instruction set
5. Developing various I/O programs for 9085, 8086 and 8051

LIST OF EXPERIMENTS
Experiment Using 8085 Microprocessor

1. Study of 8085 Microprocessor Trainer Kit
2. 8-bit Arithmetic Operations
3. (Addition, Subtraction, Multiplication and Division)
4. Block Operations
5. (Move, Exchange, Compare, Insert and Delete)
6. Code Conversions
7. Digital Clock simulation
8. Moving Display
9. Serial Communication
10. Interrupt Programming
11. Elevator Simulation
12. Traffic Light Control

Experiments Using 8086 Microprocessor with MASM

1. Arithmetic Operations
2. Sorting and Searching

Experiments Using 8051 Microcontroller

1. Arithmetic operations
2. ADC & DAC Interfacing
3. Stepper Motor and DC Motor Interface

Content beyond Syllabus:
Multi-core Programming

Websites:

1. http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0211a/index.html
2. http://www.arm.com/products/processors/classic/arm7/index.php
3. http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0211a/index.html
4. **http://www.embeddedindia.com/**

PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – P41 MICROPROCESSORS AND MICROCONTROLLERS LAB

Mini Project on Keyboard and Mouse Control

Submitted by

MOHAN SHYAME

PRAKASHRAJ . M

PRAVEEN KUMAR.E

PRAVIN KUMAR .S

PRIYADHARSHINI.M

RUBAN.O

SAGAR

SANDHIYA.R

SELVAPRIYA

SHARMILA.M

SNEHA.U

II YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

HARDWARE REQUIRED

- Arduino Leonardo, Micro or Arduino Due board
- 5 pushbuttons
- 5 10k ohm resistors
- hook-up wires
- breadboard

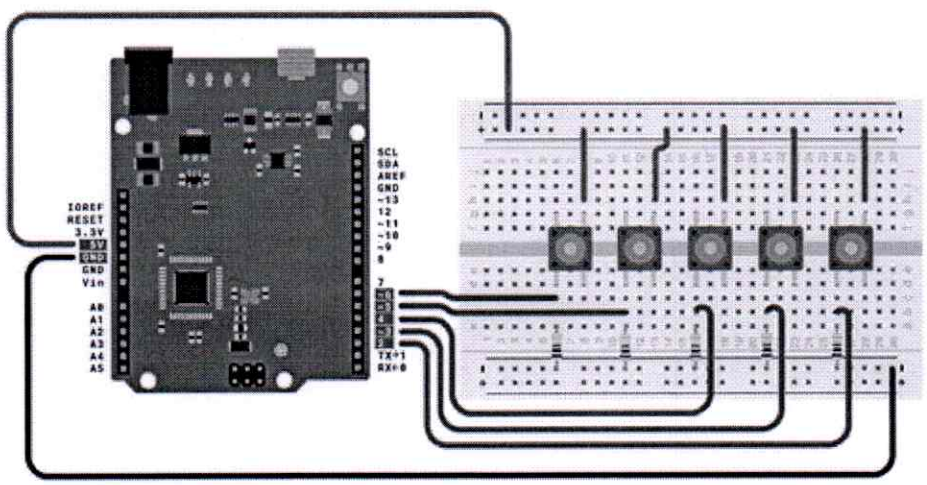
SOFTWARE REQUIRED

- Any text editor

CIRCUIT

Attach one end of the the pushbuttons to pins 2, 3, 4, 5, and 6 on the board. Attach the other end to +5V. Use the resistors as pull-downs, providing a reference to ground for the switches. Attach them from the pin connecting to the board to ground.

Once you've programmed your board, unplug the USB cable and open a text editor. Connect your board to your computer and press the buttons to write in the document as you move the cursor.



IT-P43 JAVA LAB

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P43	Java Lab	-	-	9

Pre-requisite: Object oriented programming

Course Objectives:

1. To understand the basics of java
2. To write programs in Java covering the object oriented concepts.
3. To write programs covering advanced concepts in java like thread handling, applets, RMI and JDBC

Course Outcomes: Students will learn how to write programs and develop projects in Java.

Develop Java programs to cover the following topics:

1. Simple Java program with one or more classes
2. Exception Handling
3. Inheritance
4. Packages
5. Interfaces
6. Event Handling
7. File Handling
8. Thread Handling
9. AWT controls/Java Swings/Struts framework
10. Applets
11. RMI
12. JDBC

Content beyond Syllabus:

Java's support in graphics, networking, web technology etc.

Text Books:

1. The Complete Reference Java J2SE 5th Edition, Herbert Schildt, TMH Publishing Company Ltd, NewDelhi.
2. Big Java 2nd Edition, Cay Horstmann, John Wiley and Sons.

Reference Books:

1. Java How to Program, Sixth Edition, H.M.Dietel and P.J.Dietel, Pearson Education/PHI
2. Core Java 2, Vol 1, Fundamentals, Cay.S.Horstmann and Gary Cornell, Seventh Edition, Pearson Education.
3. Core Java 2, Vol 2, Advanced Features, Cay.S.Horstmann and Gary Cornell, Seventh Edition, Pearson Education.

Websites:

1. http://www.ibm.com/developerworks/java/
2. http://docs.oracle.com/javase/tutorial/rmi/.
3. IBM's tutorials on Swings, AWT controls and JDBC.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHAL KUPPAM,
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-P43 JAVA LAB

Mini Project – Moving Balls using Java Applet

Submitted by

SRINIVASSAN

SUBALAKSHMI.S

SUBASHINI.G

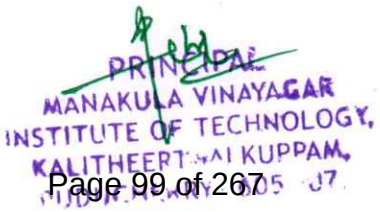
SUBHASH.B

SUDHARSAN

SUHAIB AHAMED K.A

II YEAR – IT

(2020-2021)



Write a program to develop a application in java applet which will show balls that will move across the screen .

In Moving Balls using Java Applet project we have two balls moving perpendicular to each other. Also balls have some sort of animation which will change it's color a regular intervals.

Program

```
import java.util.*;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
//Package Declarations

 /*
<applet code="balls.class" width=670 height=300>
</applet>
*/

public class balls extends Applet implements Runnable
{
    Random r = new Random(); // Random number generation for Different Colours
    int x = 10, y = 10,sang1 = 0, sang2 = 0, he = 500 , wi = 500;
    public void init()
    {
            Thread t = new Thread(this); // Thread creation for animation
            t.start();
    }
    public void run()
    {
            while(true)
            {
            try
            {
                    repaint();
                    Thread.sleep(100);  // Thread time interval set to 100 mil seconds
                    if( x < wi - 100)   // Playing with position of balls on the applet screen
                    x += 5;
                    if( y < he - 100)
```

```
                y += 5;
                if( x > wi - 100)
                x = wi - 100;
                if( y > he - 100)
                y = he - 100;
                sang1 += 10;
                sang2 += 10;
        }
        catch(Exception e)
        { }
        }
}
public void paint(Graphics g)
{
        Dimension d = getSize();
        he = d.height;
        wi = d.width;
        g.setColor(new Color(r.nextInt(255),r.nextInt(255),r.nextInt(255))); // Balls
       random colors sets
        g.fillArc(x,20,100,100,sang1,90);
        g.setColor(new Color(r.nextInt(255),r.nextInt(255),r.nextInt(255)));
        g.fillArc(x,20,100,100,sang1 + 90,90);
        g.setColor(new Color(r.nextInt(255),r.nextInt(255),r.nextInt(255)));
        g.fillArc(x,20,100,100,sang1 + 180,90);
        g.setColor(new Color(r.nextInt(255),r.nextInt(255),r.nextInt(255)));
        g.fillArc(x,20,100,100,sang1 + 270,90);
        g.setColor(new Color(r.nextInt(255),r.nextInt(255),r.nextInt(255)));
        g.fillArc(10, y, 100, 100, sang2 ,90);
        g.setColor(new Color(r.nextInt(255),r.nextInt(255),r.nextInt(255)));
        g.fillArc(10,y,100,100, sang2 + 90,90);
        g.setColor(new Color(r.nextInt(255),r.nextInt(255),r.nextInt(255)));
        g.fillArc(10,y,100,100,sang2 + 180,90);
        g.setColor(new Color(r.nextInt(255),r.nextInt(255),r.nextInt(255)));
        g.fillArc(10,y,100,100,sang2 + 270,90);
}
}
```

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM
PUDUCHERRY 605 107.

IT-T51 COMMUNICATION ENGINEERING-II

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T51	Communication Engineering- II	3	1	0

Course Objectives:

1. To learn the various orbits used for satellite communication systems.
2. To understand the working principle of various satellite systems and their applications.
3. To understand the concept of spread spectrum technologies, Rake receivers and CDMA
4. To introduce the concept and operation of cellular mobile communication and to introduce various cellular standards
5. To learn the need for fiber optics communication and the operation of fiber optic communication system.

Course Outcomes:

On successful completion of this course students will be able to:

1. Understand the operation of various types of communication systems
2. Understand the application of various types of communication systems

UNIT I **(12 Periods)**

Satellite Communication systems: Satellite Orbits, launch vehicles, look angles, satellite parameters, satellite link model and link budget calculations, satellites used for mobile networks and personal communication systems-GPS services.

UNIT II **(12Periods)**

Spread Spectrum Communication: Introduction-PN sequences-Direct sequence spread spectrum systems-Frequency hopping spread spectrum systems- slow and fast frequency hopping- RAKE receivers-principle of code division multiple access-applications.

UNIT III (12 Periods)

Cellular Mobile Communication concepts: Basic cellular concept-frequency reuse-interference-uniqueness of mobile radio environment - Performance metrics in cellular system-Elements of cellular mobile radio-Handoff-Frequency management and channel assignment-concepts of cell splitting and cell sectoring

UNIT IV **(12 Periods)**

Mobile Communication systems and standards : Second Generation systems- Global System for Mobile Communication (GSM) – architecture-channels-radio resource, mobility, communication and network management – IS-95 standard- CDMA frequency and channel allocations- CDMA traffic channels- radiated power - GPRS –architecture and services- principle of EDGE- Third Generation systems- UMTS network architecture- UTRAN architecture- basic principles of cdma2000

UNIT V **(12 Periods)**

Optical fiber communication Systems: Need for fiber optics, introduction to optical fiber, principle of light transmission through a fiber, fiber characteristics and classification, various fiber losses– Light sources and photo detectors -Block diagram of fiber optic system- Power budget analysis for an optical link-Recent applications of fiber optics.

(Total: 60 Periods)

Content beyond Syllabus:

To implement a wireless link and study its performance using computer programs

Text Books:

1. Wayne Tomasi, Electronics Communication systems, Pearson Education, Fifth edition, 2008.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
Puducherry
605 107

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

COMMUNICATION ENGINEERING-II/ IT-T51

MINI PROJECT

ON

INTERFACING GSM MODULE TO ARDUINO- SEND AND RECEIVE SMS

Submitted by

ABDUL RAHIM.S
ABDUL RAHMAN.H
ALTHAF KHAN. J
AMIRTHALAKSHMI.P
ANJANA.V
ARUNA.G

III YEAR (2020-2021)

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY * Puducherry 605 107

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

Miniproject on Interfacing GSM Module to Arduino-Send and Receive SMS

Aim:

To interface GSM module to arduino for sending and receiving SMS

Software required:

MATLAB Software, PC

Objective:

In this article, interfacing **GSM Module to Arduino** was explained. There are different kinds of GSM modules available in market. Here, the most popular module based on **Simcom SIM900** and **Arduino Uno** was used. Interfacing a GSM module to Arduino is pretty simple with 3 connections between the GSM module and arduino.

A **GSM Module** is basically a GSM Modem (like SIM 900) connected to a PCB with different types of output taken from the board – say TTL Output (for Arduino, 8051 and other microcontrollers) and RS232 Output to interface directly with a PC (personal computer). The board will also have pins or provisions to attach mic and speaker, to take out +5V or other values of power and ground connections. These type of provisions vary with different modules.

IT-T52 SOFTWARE ENGINEERING

Subject Code	Subject Name	Lectures (Periods)	Tutorial (Periods)	Practical (Periods)
IT-T52	Software Engineering	3	1	0

Course Objectives:

1. To learn, practice and apply the software engineering industry practices.
2. To acquire knowledge on the various techniques, tools and models for each of the phases of software development.

Course Outcomes:

1. Ability to apply basic knowledge and understanding of the analysis, synthesis and design of complex systems
2. Develop, maintain and evaluate large-scale software systems
3. Produce efficient, reliable, robust and cost-effective software solutions

Syllabus:

UNIT I

Introduction to Software Engineering: The Software Engineering Discipline – Evolution and Impact – Software Development projects – Emergence of Software Engineering – Computer System Engineering **Software Life Cycle Models:** classic Waterfall model – Iterative Lifecycle model – prototyping model – Evolutionary model – spiral model – Comparison of Life cycle models.

UNIT II

Software Project Management: Responsibilities of a Software Project Manager – Project Planning – Metrics for Project Size Estimation – Empirical Estimation Techniques – COCOMO – Halstead's Software Science – Staffing Level Estimation – Scheduling – Organization and Team structures – Staffing – Risk Management – Software Configuration Management

Requirements Analysis and Specification: Requirements Gathering and Analysis – Software Requirements specification – Formal System Specification – Axiomatic Specification - Algebraic Specification – 4GL.

UNIT III

Software Design: Outcome of a Design Process – Characteristiscs of a Good Software Design – Coupling and Cohesion – Approaches to Software Design – Object Oriented Vs Function Oriented Software Design approaches

Function Oriented Software Design: Structured Analysis – Data Flow Diagrams – Applying DFD to Real time systems – Structured and Detailed Design.

UNIT IV

Coding and Testing: Coding – Software Documentation – Testing – Unit Testing – Black Box testing – White Box testing – Debugging – Program Analysis tools – Integration testing – Testing Object Oriented programs – System Testing – Issues.

UNIT V

Software Reliability and Quality : - Software Reliability – Software Quality – ISO 9000 – SEI CMM – Six Sigma.

CASE and Software Maintenance: - CASE environment – CASE support in Software Life cycle – Characteristics of CASE tools – characteristics of software maintenance – software reverse engineering – software maintenance process models.

(Total: 60 Periods)

Content beyond the Syllabus:

The students can be encouraged to apply concepts learnt in this course for the development / documentation of their mini project and final year project

Text Books:

1. Rajib Mall, " Fundamentals of Software Engineering", PHI Learning, Third Edition, 2013.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – T52 SOFTWARE ENGINEERING

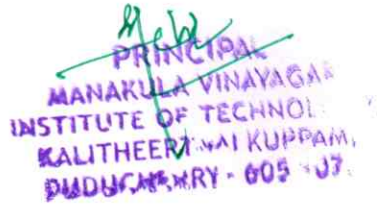
MINI PROJECT FOR ELECTRONIC CASH COUNTER

Submitted by

ATHIFFA.M.A
AUROBINDH SAI.C
BALAMURUGAN.P
DEEPIKA.S
DEEPIKA.V
DHIVAGAR.K

III YEAR – IT

(2020-2021)



ELECTRONIC CASH COUNTER

This project is mainly developed for the **Account** Division of a Banking sector to provide better **interface of** the entire **banking transactions**. This system is aimed to give a better out look to the user interfaces and to implement all the banking transactions:

Some operations include in the project are:

- ✓ **Supply of Account Information**
- ✓ **New Account Creations**
- ✓ **Deposits**
- ✓ **Withdraws**
- ✓ **Cheque book issues**
- ✓ **Stop payments**
- ✓ **Transfer of accounts**
- ✓ **Report**

Generations. Proposed

System:

The development of the new system contains the following activities, which try to automate

the entire process keeping in view of the database integration approach.

- ➢ User friendliness is provided in the application with various controls.
- ➢ The system makes the overall project management much easier and flexible.
- ➢ Readily upload the latest updates, allows user to download the alerts by clicking the URL.
- ➢ There is no risk of data mismanagement at any level while the project development is under process.
- ➢ It provides high level of security with different level of authentication

Software Requirement Analysis:

Actors Involved in Electronic Cash Payment Systems:

- ➢ **Customers:** Customers use the digital cash payment systems to make purchases.
- ➢ **Dealers:** Dealers have to bear the costs of payment transactions.
- ➢ **Providers for digital payment systems:** Providers are intermediaries between dealers and financial institutions. They provide services and training.
- ➢ **Financial institutions:** Banking systems or organizations that use electronic payment systems.
- ➢ **Trust Centers:** They control digital signature keys, and help to secure customer confidence in certain payment systems. They are responsible for the integrity of transmitted data and authenticity of contractors.

Customer:

Functional requirements:

- Create new accounts.
- Provide account information.
- Money transferring to other users.
- Give multiple payment options. Ex: cash pay, wallets pay, coupons pay, cheque pay.
- Getting the reports of transactions.
- Transaction status report.

Non functional requirements:

Security

- Transaction data must be encrypted. So that no one can access the account information.
- Not to allow user to check the other users account data.

Reliability

- Unsuccessful transaction must recovery immediately.
- Make sure no data loss.

Portability

- Able to do transaction anywhere.

Usability

- Easy to understand and easy to use.

Dealers:

Functional requirements:

- Provide customer contact data and transaction details. (not complete information)
- Getting the reports of transactions.
- Can create account.
- Transaction status report.

Non functional requirements:

Security

- Bank account data hiding.

IT-T53 OPERATING SYSTEMS

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T53	Operating Systems	3	1	0

Course Objectives:

1. To grasp a fundamental understanding of operating systems
2. To learn the concepts and creation computer processes and threads
3. To understand memory management and virtual memory concepts in modern OS
4. To understand process concurrency and synchronization
5. Understand the concepts of data input/output, storage and file management
6. To learn the scheduling policies, memory management and file management of some commercial operating systems

Course Outcomes:

On successful completion of this course students will be able to:

1. Understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.,
2. Understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions,
3. Understand how the operating system abstractions can be implemented,
4. Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software,
5. Understand basic resource management techniques (scheduling or time management, space management) and principles and how they can be implemented. These also include issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection.

Unit I **(12 Periods)**

Introduction: Mainframe Systems – Desktop Systems – Multiprocessor Systems – Distributed Systems – Clustered Systems – Real Time Systems – Hardware Protection – System Components – Handheld Systems - Operating System Services – System Calls – System Programs – Process Concept – Process Scheduling – Operations on Processes – Cooperating Processes – Inter-process Communication.

Unit II **(12 Periods)**

Threads – Overview–Threading issues-CPU Scheduling–Basic Concepts–Scheduling Criteria – Scheduling Algorithms–Multiple – Processor Scheduling – Real Time Scheduling – The Critical- SectionProblem – Synchronization Hardware-Semaphores– Classic problems of Synchronization –Critical regions–Monitors.

Unit III **(12 Periods)**

System Model – Deadlock Characterization – Methods for handling Deadlocks –Deadlock Prevention – Deadlock avoidance – Deadlock detection – Recovery from Deadlocks – Storage Management –Swapping – Contiguous Memory allocation – Paging – Segmentation – Segmentation with Paging -Virtual Memory – Demand Paging – Process creation – Page Replacement – Allocation of frames – Thrashing.

Unit IV **(12 Periods)**

File Concept – Access Methods–Directory Structure–File System Mounting–File Sharing–Protection -File System Structure – File System Implementation – Directory Implementation – Allocation Methods - Free-space Management-Kernel I/O Subsystems – Disk Structure – Disk Scheduling – Disk Management – Swap-Space Management.

Unit V **(12 Periods)**

Linux overview – Kernel Architecture–Process, memory, file and I/O management –Inter- process communication and synchronization – Security - Windows XP - System Architecture – System management mechanisms – Process, thread, memory and file management– I/O subsystem–Interprocess communication–Security

(Total : 60 Periods)

Content beyond Syllabus:

1. Introduction to Multiprocessor, Network and Distributed Operating Systems.

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, Seventh Edition, John Wiley & Sons (ASIA)Pvt.Ltd,2005.
2. Harvey M. Deitel, PaulJ. Deitel, andDavid R. Choffnes, Operating Systems, Third Edition,Prentice Hall,2003.

Reference Books:

1. William Stallings, Operating System, Prentice Hall of India, 6th Edition, 2009.
2. Harvey M. Deitel, Operating System, Second Edition, Pearson Education Pvt. Ltd, 2002.
3. Gary J. Nutt, Operating Systems: A Modern Perspective, Second Edition, Addison Wesley, 2001.

Websites:

1. http://www.tcyonline.com/tests/operating-system-concepts
2. http://www.galvin.info/history-of-operating-system-concepts-textbook
3. http://www.ittestpapers.com/operating-system-concepts

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-T53 OPERATING SYSTEM

Mini Project on Bankers Algorithm for Deadlock Avoidance

Submitted by

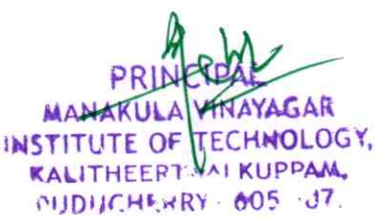
DIVYA.A

ESHWARAMOORTHY.S A

HARIDHA.S

III YEAR – IT

(2020-2021)



Bankers Algorithm for Deadlock Avoidance

Objective:

Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance.

Procedure:

In a multiprogramming environment, several processes may compete for a finite number of resources. A process requests resources; if the resources are not available at that time, the process enters a waiting state. Sometimes, a waiting process is never again able to change state, because the resources it has requested are held by other waiting processes. This situation is called a deadlock. Deadlock avoidance is one of the techniques for handling deadlocks. This approach requires that the operating system be given in advance additional information concerning which resources a process will request and use during its lifetime. With this additional knowledge, it can decide for each request whether or not the process should wait. To decide whether the current request can be satisfied or must be delayed, the system must consider the resources currently available, the resources currently allocated to each process, and the future requests and releases of each process. Banker's algorithm is a deadlock avoidance algorithm that is applicable to a system with multiple instances of each resource type.

Program:

```
#include<stdio.h>
struct file
{
        int all[10];
        int max[10];
        int need[10];
        int flag;
};
void main()
{
        struct file f[10];
        int fl;
        int i,j,k,p,b,n,r,g,cnt=0,id,newr;
        int avail[10],seq[10];
        clrscr();
        printf("Enter number of processes--");
        scanf("%d",&n);
        printf("Enter number of resources--");
        scanf("%d",&r);
        for(i=0;i<n;i++)
        {
                printf("Enter details for P%d",i);
                printf("\nEnter allocation\t--\t");
                for(j=0;j<r;j++)
                        scanf("%d",&f[i].all[j]);
                        printf("Enter Max\t\t -- \t");
```

IT-T54 DATABASE MANAGEMENT SYSTEMS

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T54	Data Base Management Systems	3	1	0

Pre-requisite:

1 Knowledge about Memory Management
2 Programming Skills

Course Objectives:

To introduce the fundamental concepts of Database Management System to the students and to make them understand the usage of Database Management System in the current industry scenario.

Course Outcomes:

The students can be able to understand the concepts of Database Management System and to choose and design the database for the specific requirement of the project

Unit I

Introduction: Introduction to Database Systems: Overview – Data Models – Database System Architecture – History of Database Systems. Entity-Relationship Model: Basic Concepts – Constraints – Keys – Design Issues – Entity Relationship Diagram – Weak Entity Sets – Extended E-R Features – Design of an E-R Database Schema.

Unit II

Relational Model: Structure of Relational Databases – Relational Algebra – Extended - Relational Algebra Operations – Modification of Database – Views – Tuple Relational - Calculus – Domain Relational Calculus. SQL: Background – Basic Structure – Set - Operations – Aggregate Functions – Null Values – Nested Sub-queries – Views – Complex Queries – Modification of the database –Joined Relations – Data-Definition Language.

Unit III

Integrity and Security: Domain Constraints – Referential Integrity – Assertions –Triggers – Security and Authorization – Authorization in SQL .Relational-Database Design: Normalization -First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form.

Unit IV

Storage and File Structures: Overview of Physical Storage Media – Magnetic Disks – RAID – Tertiary Storage – Storage Access – File Organization. Indexing and Hashing: Basic Concepts –Static Hashing – Dynamic Hashing.

Unit V

Transactions: Transaction concept – Transaction State – Implementation of Atomicity and Durability – Concurrent Executions – Serializability – Testing for Serializability. Concurrency Control: Lock-Based Protocols – Timestamp-Based Protocols. Recovery System: Failure Classification – Storage Structure – Recovery and Atomicity – Log-Based Recovery – Shadow Paging.

(Total : 45 Periods)

Content beyond Syllabus: The recent developments of the Database Management System and the current standards of the IT organizations have to be introduced to the students.

Text Books:

1. Silberschatz, Korth, Sudarshan, *Database System Concepts*, 6th Edition – McGraw-Hill Higher Education, International Edition, 2011.

Reference Books:

1. Fred R McFadden, Jeffery A Hoffer, Mary B. Prescott, Modern Database Management, Seventh Edition, Addison Wesley, 2004.
2. Elmasri, Navathe, Fundamentals of database Systems, Sixth Edition, Addison Wesley, 2010.
3. JefreyD.Ulman, Jenifer Widom, A First Course in Database Systems, Pearson Education Asia, 2001.
4. Bipin C Desai, An Introduction to Database Systems, Galgotia Publications Pvt Limited, 2003.

Websites:

1. http://www.database.com/
2. www.infoworld.com/t/dbms

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY Puducherry 605 107

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT T54 Data Base Management Systems

Mini Project on Employee Management System

Submitted by

HARSHA.K.C

III YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
Puducherry
605 107

AIM

To develop a software to maintain the Employee details for an organization using JDBC connectivity.

ALGORITHM

STEP 1: Start

STEP 2: Create the Emp1 table with its essential attributes(Emp1(Eno, Ename,salary))

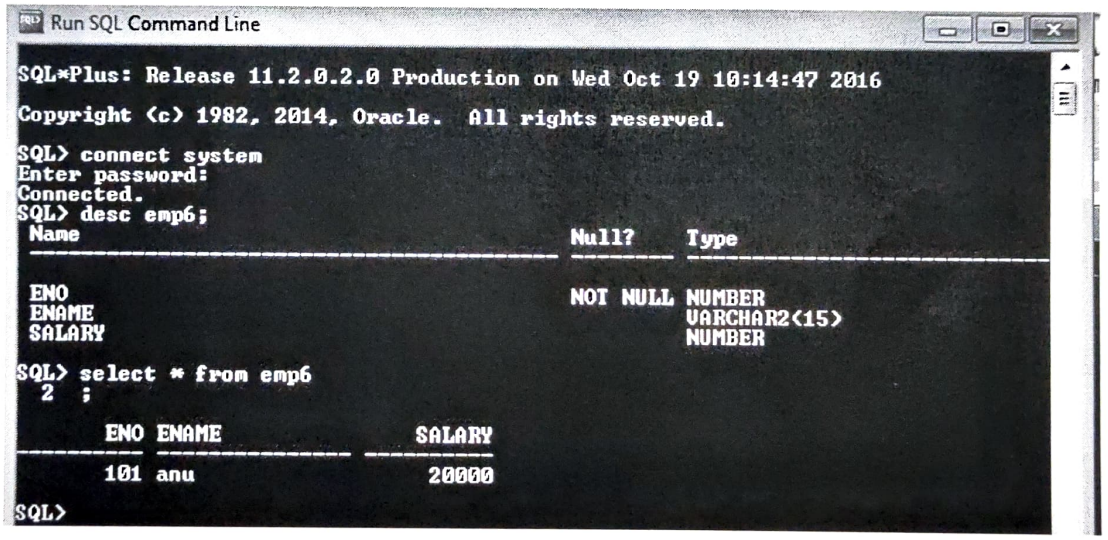
STEP 3: Insert attribute values into the emp1 table using jdbc connectivity

STEP 4: Update the attribute values into the emp1 table using jdbc connectivity

STEP 5: Delete the attribute values into the emp1 table using jdbc connectivity

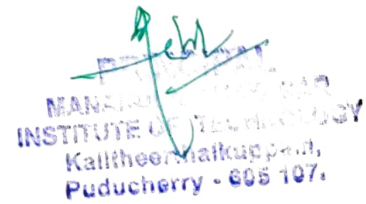
STEP 6: Get the information of employee using select queries from the emp1 table using jdbc connectivity

STEP 5: Stop



Insert.java

```
package javaapplication2;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.*;
import java.util.Scanner;
class Insert{
public static void main(String args[]){
try{
    Scanner s=new Scanner(System.in);
    System.out.println("Insert the Values in Table");
    System.out.println("Enter the Employee No:");
    int eno=s.nextInt();
    System.out.println("Enter the Employee Name:");
    String name=s.next();
    System.out.println("Enter the salary:");
    int salary=s.nextInt();
//step1 load the driver class
Class.forName("oracle.jdbc.driver.OracleDriver");

//step2 create  the connection object
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","system");
```



```
 //step3 create the statement object
//Statement stmt=con.createStatement();
PreparedStatement ps;
ps=con.prepareStatement("insert into empl(eno,name,salary) values(?,?,?)");
ps.setInt(1, eno);
ps.setString(2,name);
ps.setInt(3,salary);
ps.executeUpdate();
System.out.println("Details Entered Successfully");
con.close();
catch(Exception e)
{
   System.out.println(e);       }       }       }
```

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
Puducherry
605 107

Delete.java

```
package javaapplication2;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.*;
import java.util.Scanner;
class delete{
public static void main(String args[]){
try{
   Scanner s=new Scanner(System.in);
   System.out.println("enter the employee no");
     int eno=s.nextInt();
    //step1 load the driver class
Class.forName("oracle.jdbc.driver.OracleDriver");

//step2 create  the connection object
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","system");

//step3 create the statement object
//Statement stmt=con.createStatement();
PreparedStatement ps;
ps=con.prepareStatement("delete from empl where eno= "+ eno + "");
ps.executeUpdate();
System.out.println("Details Entered Successfully");
```

Get.java

```
package javaapplication2;
import java.sql.*;
class get{
public static void main(String args[]){
try{
//step1 load the driver class
Class.forName("oracle.jdbc.driver.OracleDriver");
 //step2 create  the connection object
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","system");
  //step3 create the statement object
Statement stmt=con.createStatement();
  //step4 execute query
ResultSet rs=stmt.executeQuery("select * from empl");
System.out.println("Eno Name    Salary");
while(rs.next())
System.out.println(+rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getInt(3));
  // System.out.println(eno + "  " + name+"   "+salary);
//step5 close the connection object
con.close();

}catch(Exception e){ System.out.println(e);}
```

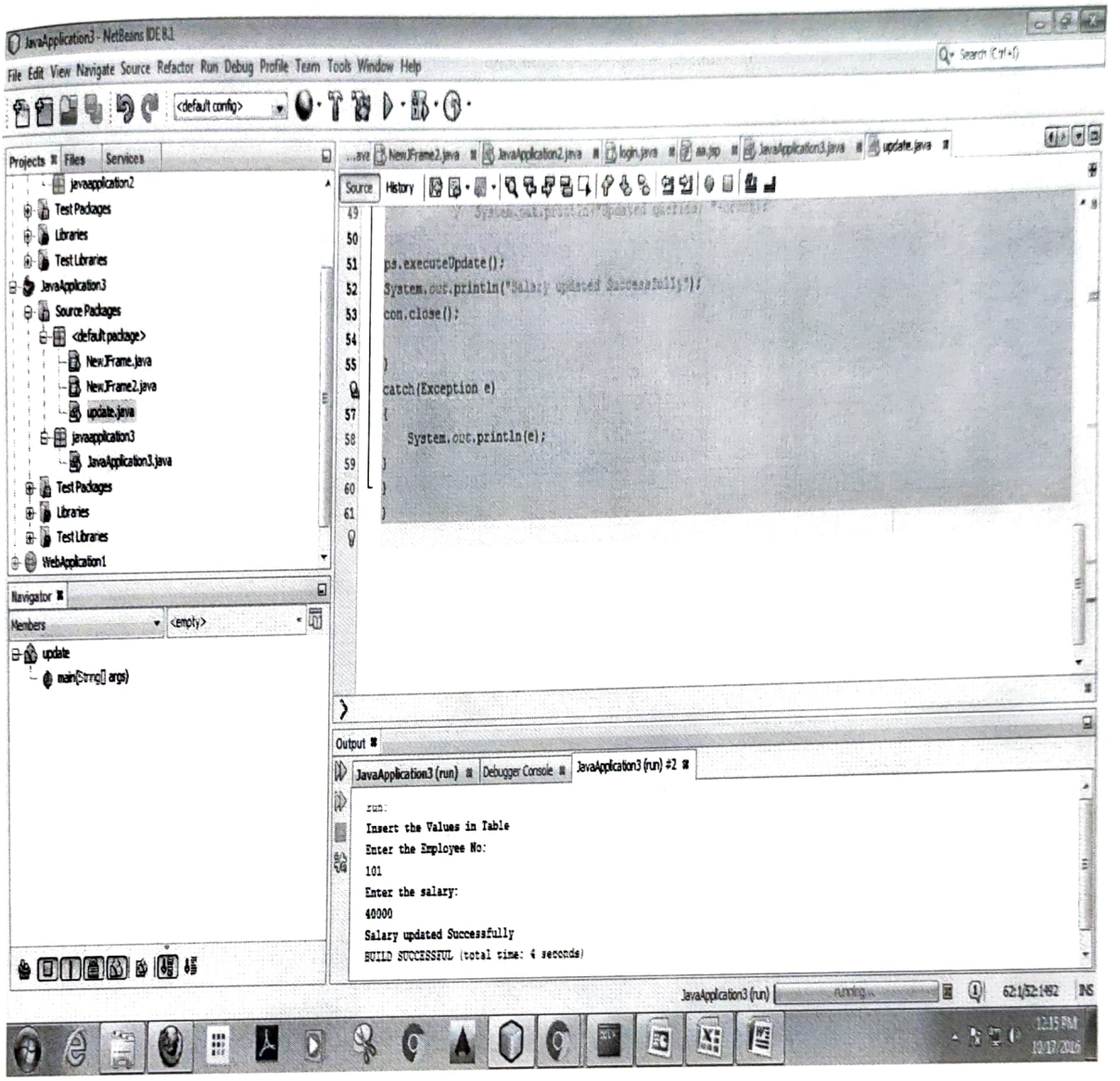
Update.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.*;
import java.util.Scanner;
public class update {

 public static void main(String args[])
 {
 try{
    Scanner s=new Scanner(System.in);
   System.out.println("Insert the Values in Table");
    System.out.println("Enter the Employee No:");
    int eno=s.nextInt();
   System.out.println("Enter the salary:");
   int salary=s.nextInt();
//step1 load the driver class
Class.forName("oracle.jdbc.driver.OracleDriver");
```

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
Puducherry
605 107

```
//step2 create  the connection object
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","it","it");

//step3 create the statement object
//Statement stmt=con.createStatement();
PreparedStatement ps;
ps=con.prepareStatement("update emp6 set salary=? where eno=?");
ps.setInt(1, salary);
ps.setInt(2,eno);
ps.executeUpdate();
System.out.println("Salary updated Successfully");
con.close();
  }
catch(Exception e)
{
   System.out.println(e);
}       }       }
```



RESULT

Thus the schema diagram for Employee details was studied and the queries are executed successfully.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY Puducherry 605 107

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthaikuppam,
Puducherry - 605 107.

IT-T55 THEORY OF COMPUTATION

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T55	Theory of Computation	3	1	-

Pre-requisite:
Knowledge in mathematics, including a course in Discrete mathematics, and in programming.

Course Objectives:
Learning about automata, grammar, language, and their relationships. Further, gives an understanding of the power of Turing machine, and the decidable nature of a problem. Also, gives the idea on some new trends and applications.

Course Outcomes:
Upon completion of the course, you should be able to:

1. Explain the basic concepts of deterministic and non-deterministic finite automata, regular language, context-free language, Turing machines, computability and complexity.
2. Describe the formal relationships among machines, languages and grammars.
3. Solve the problems using formal language.
4. Develop a view on the importance of computational theory.

Unit I
Finite Automata and Regular Expressions: Deterministic and Non-Deterministic Finite Automata,
Finite Automata with ε-moves, regular expressions – equivalence of NFA and DFA, two-way finite automata, Moore and Mealy machines, minimization of finite automata- applications of finite automata.

Unit II
Regular Expressions and Context Free Grammars: Regular expression formalism- equivalence with finite automata-regular sets and closure properties- pumping lemma for regular languages- decision algorithms for regular sets- applications.
Context-Free Grammars – derivation trees, Chomsky Normal Forms and Greibach Normal Forms, ambiguous and unambiguous grammars- equivalence of regular grammar and finite automata- pumping lemma for Context free languages – applications.

Unit III
Turing machines: Elements of Turing machines(TM) – tuning machine construction – complexity of TM-Universal, multi-tape, multi-track, multi-stack Turing machines –recursive and recursive enumerable languages- functions –Church's Turing hypothesis.

Unit IV
Pushdown Automata(PDA) and Parsing Algorithms: Pushdown Automata and context-free languages; Deterministic PDA-Nondeterministic PDA- Equivalence of PDA and CFG-closure properties of CFL.

Unit V
Parsing Techniques: Top down parsing- bottom up parsing- Automatic construction of bottom up parsers – LR(0) grammar – SLR parser – LR(1) grammar – Canonical-LR parser- LALR parser.

(Total :60 Periods)

Content beyond Syllabus:

1. Models of Linear bounded automata
2. Partial recursive functions and Turing machines.

Text Books:

1. Vivek Kulkarni, "Theory of Computation", Oxford University press, 2013.
2. K.L.P. Mishra and NM.Chandrasekaran, "Theory of Computer Science-Automata Languages and Computation", third edition, PHI Learning Private Ltd, 2009.
3. John E. Hopcroft and Jeffrey D. Ullman, Introduction to Automata Theory, Languages and Computation, Narosa Publishers, 2002.

Reference Books:

1. Michael Sipser, Introduction to the Theory of Computations, Brooks/Cole Thomson Learning, 1997.
2. John c. Martin, Introduction to Languages and the Theory of Computation, Tata McGraw-Hill,2003.

Websites:

1. www.infolab.stanford.edu/~ullman/ialc.html
2. www.nptel.iitm.ac.in/courses/106106049/

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT T55 Theory of Computation

Mini Project on Design of Turing Machine

Submitted by

LAVANYA.V

MAHALAKSHMI.S

MAHALAKSHMI.S

MAHESWARI.S

NAGARAJ

PARTHIBAN.R

III YEAR – IT

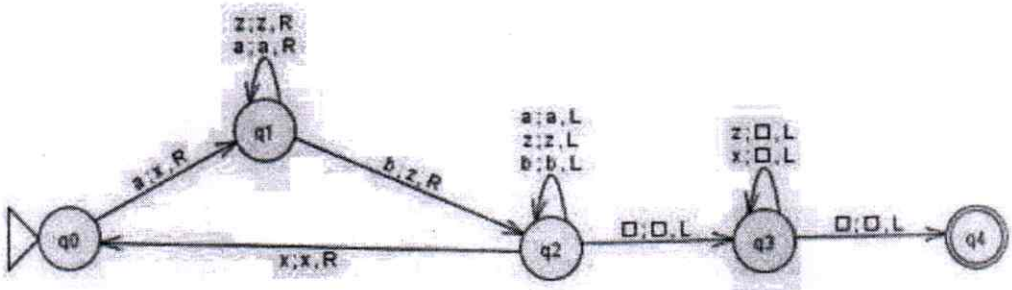
(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

Aim: To Construct a Turing machine that accepts the following languages.

a) L = {an bn | where n > 0}, Σ = {a,b}

Solution:



Result: Thus Constructed the Turing machine that accepts the given languages.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

IT-E51 COMPUTER HARDWARE AND TROUBLESHOOTING

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-E51	Computer Hardware and Troubleshooting	3	1	0

Course Objectives:

1. It provides insight to the various parts and types of computer.
2. It familiarizes the hardware types and the evolution in each of them.
3. It also gives the basics of troubleshooting.

Course Outcomes:

On successful completion of this course students will be able to:

The students will have theoretical exposure as well as hands on exposure to know about the hardware aspects of computer.

Unit I

PC Hardware Overview: Introduction–BasicPartsofPC–Functionalblockdiagram–systemboard–Microprocessor–Interrupts– DMA–SMPS–BIOS–POSTsequence-Systemconfigurationion–Memory–Massstorage–I/Ointerfacestandards.

Unit II

Bus Standards and Networking: ISA–PCI–SCSI–IDE–USB–comparative study and characteristics–Network Interface Cards–Cables and connectors-MODEM–AT command set.

Unit III

Peripheral Devices & Display Adapters: Functional descriptions of keyboard – mouse – printers – joystick – scanners – CGA – SVGA.

Unit IV

Mass Storage Devices: Floppy disk and drive – Hard disk and drive – MFM and RLL recording standards – CD Technology – DVD technology – pen drives – tape drives.

Unit V

Troubleshooting tools: In-Circuit Emulators – Logic State / Timing Analyzers – Digital Multi meters – CROs–Signature Analyzers– Troubleshooting problems of system boards, add on cards and peripherals.

Content beyond Syllabus:

1. Advanced data structures and their implementation
2. Implementation of the data structures in different language platforms

Text Books:

1. Hans Peter Messmer, Indispensable PC Hardware Book , Pearson Education, 4th edition,2003.
2. Govindarajulu, IBM PC and Clones , Tata McGraw Hill, 4th edition, 2002.

Reference Books:

1. Barry Brey, The Intel Microprocessors 8086/88, 80186/188, 80286, 80386,80486, PENTIUM and PENTIUM PRO architecture, Programming and Interfacing, 6th edition, PHI, 2002.
2. Ed Tittel, David Johnson, Networking Essentials: Study Guide, Comdex Computer Publishing,1998.
3. Scott Muller, Upgrading and Repairing PCs, 15th edition, 2002.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – E51 COMPUTER HARDWARE AND TROUBLESHOOTING

Mini Project on Hard Disk Partitioning

Submitted by

PRIYADARSHINI.M

PRIYADHARSHINI.S

RAGHUL.T.P

RENGARAJAN.T.R

RISHIKA K

SAMPATHKUMAR D

III YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107

AIM:

To partition the given hard disk into four drives such as the first drive C, second drive D, third drive E and fourth drive F of capacity 5GB each.

REQUIREMENTS:

1. PC with CD-ROM
2. Bootable CD with fdisk utility
3. Hard disk of capacity 20GB

THEORY:

Disk partition is creation of logical drive upon a hard disk operating system allow users to divide a hard disk into multiple partitions. Partitioning a hard disk drive defines specific areas within the disk. A partition may constitute an entire logical drive are it may form a part of large virtual drive which could span over several partition and hard disk.

Partitioning makes it possible to create several file system (either of same type or of different) on single hard disk. Some of the commonly used file systems are

1. FAT (File Allocation Table)
2. VFAT (Virtual File Allocation Table)
3. NTFS (new Technology File System)
4. FAT 32 (32 bit File Allocation Table)

A hard disk would contain either as many as four primary partitions or one to three partitions along with a single extended partition. Each of these partitions are described by a 16-byte entry in the partition table which is loaded in the partition table which is loaded in the master boot record the type 'A' partition is identified by a byte code found in partition table entry. Most of the operating system use "fdisk" command to create hard disk partitions.

PRIMARY PARTITION:

A primary (or logical) partition contains one file system, it may be NTFS or FAT file system. The first partition (c :) must be a primary partition.

EXTENDED PARTITION:

An extended partition is secondary to the primary partition. A hard disk may contain only one extended partition, which can then be sub-divided into logical drives each of which

(under dos and windows) assigned additional drive letters. Extended partition is useful if you want more than four partitions on a single physical drive.

MICROSOFT WINDOWS PARTITION SCHEME:

With Microsoft windows, the standard partitioning scheme is to create a single active primary partition, the c: drive, where the operation system user data, applications and page fill all reside. Some users how're, prefer to create multiple partition so that the operating system can be stored separately from other kind of data.

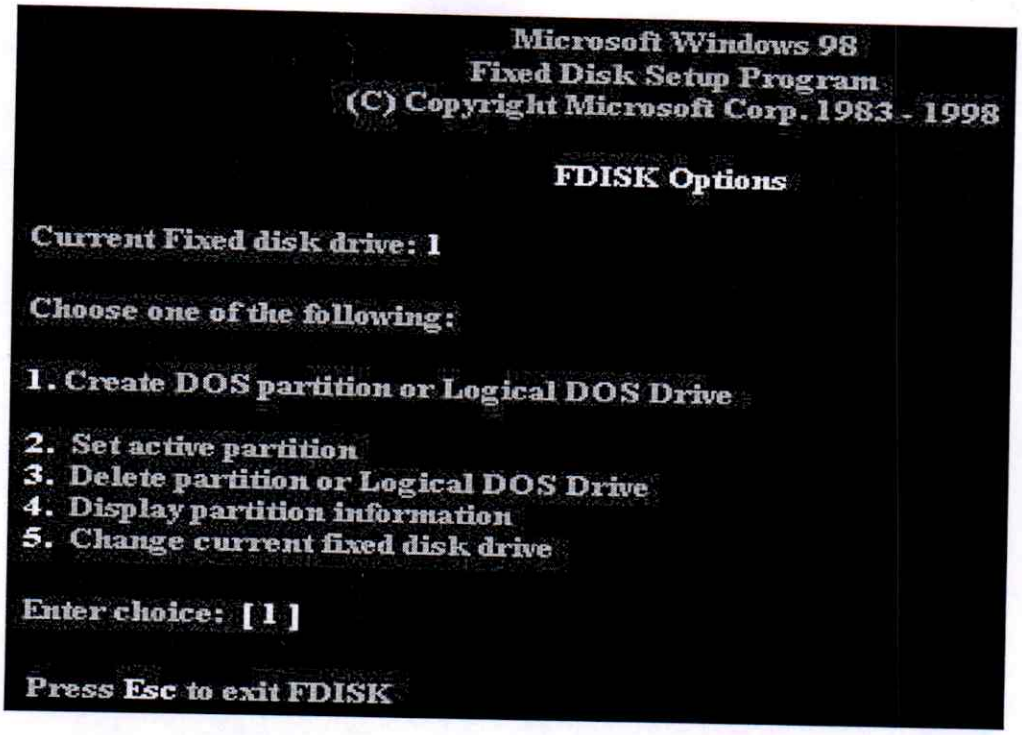
UNIX PARTITION:

For Unix-based and Unix-like operating system such as Linux and Mac os x, the creation of separate partition for lboot,lhome,ltmp,luser,lvar,swap and all remaining file under the "/" (root directory) is possible.

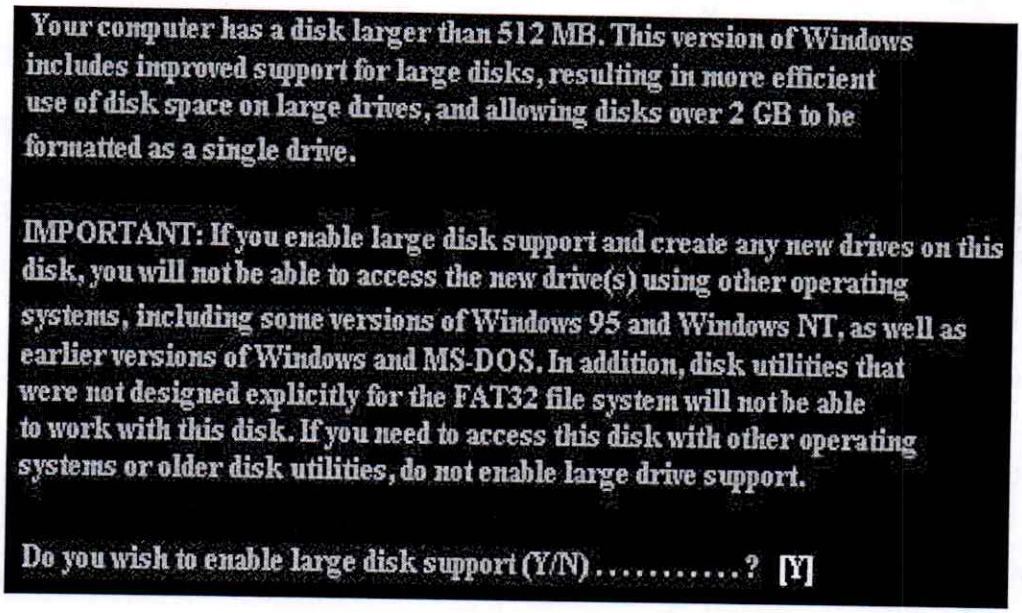
HARD DISK PARTITIONS:

This procedure explains how to setup a **new hard disk.** Before a new hard disk can be used it needs to be setup. This involves partitioning and formatting the hard disk. Windows 98 or ME boot disk contains the required software to perform this procedure. FDISK.EXE and FORMAT.COM are the files required in your bootable floppy disk.

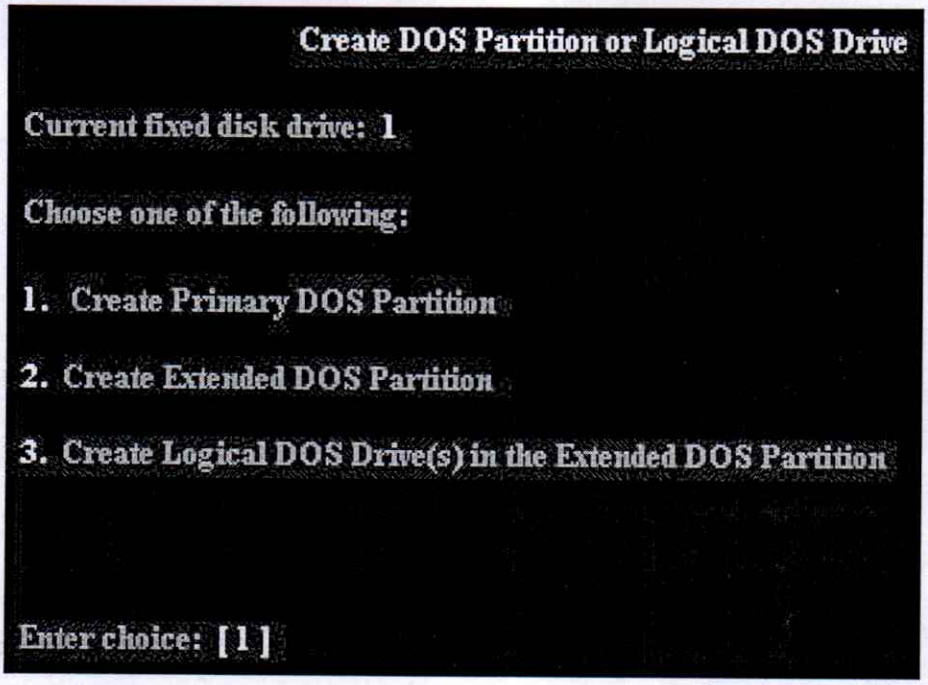
Start the partition and format procedure by booting your PC using a Windows boot disk. Make sure you set the BIOS so that the boot sequence is set to detect the floppy disk first. If your system has no problems booting you will be presented with a Windows boot disk menu. This gives you the option to start the system with or without CD-ROM support. At this stage you do not need the CD-ROM support, so choose the option to boot without CD-ROM support. You should end up in the MS DOS prompt A: (A drive). From A: command prompt type **fdisk.** You will be presented with following message:



Choose "Y" to enable large disk support. You will now be presented with the FDISK main menu as shown below:



From the menu, choose option 1 - Create DOS partition or Logical DOS drive. Another menu will present the following options.



Choose option 1 - Create primary DOS Partition. FDISK verifies the integrity of your drive and will ask you if want to use the maximum available size of your hard disk to create the primary partition and set it active. To keep things simple we will create one large partition. Choose "Y" to use maximum available space.

When the partition has been created successfully you will be notified by the system. Your drive is now known as C: (C drive). Press "Esc" to return to the menu. Press "Esc" again to exit FDISK. You need to restart your system for the changes to take effect. Leave boot disk in the drive. When the system reboots, choose start without CD-ROM from the boot disk menu. While booting from floppy disk you might get error message like "Invalid media type reading drive C" this is OK for this stage as the hard disk is not formatted. If you want to create extended DOS partitions, specify the size of the partitions in the bytes for primary partition. And remaining space can be utilized to create logical drives in the extended partition.

RESULT:

The given hard disk is partitioned according to the requirements.

IT-P51 COMMUNICATION ENGINEERING LAB

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P51	Communication Engineering Lab	0	0	3

Course Objectives:

1. To understand the working of main concepts of analogue and digital communication systems.
2. To enhance technical skills through analyzing the waveforms obtained at various stages of the experiment.
3. To verify the experimentally obtained and simulated outputs and knowing the reason for the deviation.

Course Outcomes:

On successful completion of this course students will be able to:

1. Follow rapid developments in the field of communication systems.
2. Apply problem-solving skills, Recognize and utilize latest analogue and digital communication technologies.
3. Interpret and integrate diverse information sources to form a coherent understanding of the subject.

Syllabus:

1. Amplitude modulation and demodulation
2. Frequency modulation and demodulation
3. PCM encoder and decoder
4. Generation of PAM, PWM and PPM
5. Generation of ASK, FSK and PSK
6. Simulation analysis of hand off performance in cellular mobile systems
7. Simulation of satellite link budget analysis
8. Simulation of fiber optic link budget analysis
9. Simulation of various propagation models (Outdoor and Indoor)
10. Simulation of antenna radiation pattern(Horn, Parabolic reflector)

Content beyond Syllabus:

1. Students will be motivated to visit the websites of AIR and Doordharshan and understand the practical frequency assignment, broadcast power level and coverage area of all the Indian radio and TV stations.

Websites:

1. http://drdo.gov.in/drdo/labs/LRDE/English

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
Puducherry
605 107

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY
COMMUNICATION ENGINEERING LABORATORY/ IT-P51

MINI PROJECT

ON

Sending SMS using GSM module and PIC microcontroller

III YEAR

(2020-2021)

Submitted by,

SANDHIYA.V
SANKARI.S
SANTHIYA.M
SARAN.P
SARANYA.S
SARANYADHEVI.S

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY * Puducherry 605 107

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM
PUDUCHERRY - 605 107.

Mini project on sending SMS using GSM module and PIC microcontroller

Aim:

To send SMS using module and PIC Microcontroller.

Apparatus Required:

PIC Microcontroller, GSM Module,

Objective:

Send SMS using GSM module and **PIC microcontroller**: This article explains how to send using **module** and PIC microcontroller. What is **AT commands** and their use to send and **receive** .

What are AT command?

AT commands are used to initialize whatever you want module to do for you. SIMCOM a company of sim tech have a AT command complete document for SIM900. You can easily download it from google. For each SIM900D feature there are separate AT command available. User can easily use these commands to configure whatever functionality they want to use from module available features. In this article I will discuss only AT commands use to send . But you will get an idea how to use rest of AT commands for using other features of module. AT commands you need to know for sending through module SIM900D and how to use while writing code:

For example we want to send "Welcome to microcontrollerslab.com".

IT-P52 OPERATING SYSTEMS LAB

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P52	Operating Systems Lab	0	0	3

Course Objectives:

1. To simulate the scheduling algorithms
2. To implement dining philophers, reader-writer's using synchronization mechanisms.
3. To learn the concept of memory management and file systems.

Course Outcomes:

On successful completion of this course students will be able to:

1. Learn the concepts of job scheduling in systems.
2. Learn event synchronization mechanisms.
3. Study the concept of memory management.

Syllabus:

1. Study of basic Unix/Linux commands.
2. Shell Programming.
3. Programs using the following systemcalls of Unix/Linux operating system:

   fork, exec, getpid, exit, wait, close, stat,opendir, readdir
4. ProgramsusingtheI/OsystemcallsofUNIXoperatingsystem(open,read,write,etc).
5. Simulations of Unix/Linux commands like ls, grep, etc.
6. Simulation of scheduling algorithms (CPUandDisk).
7. Implementation of synchronization problems using Semaphore.
8. Simulation of basic memory management schemes.
9. Simulation of virtual memory management schemes.
10. Simulation of filesystems.

Content beyond Syllabus:

1. Simulation environment for deadlock situation can be created.
2. Paging and segmentation may be included.

Websites:

1. http://www.inf.ed.ac.uk/teaching/courses/os/prac/
2. http://www.scribd.com/doc/7137624/OS-Practical-File/
3. http://www.cl.cam.ac.uk/freshers/raspberrypi/tutorials/os/introduction.html/

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHAL KUPPAM,
PUDUCHERRY - 605 107

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – P62 OPERATING SYSTEM LAB

MEMORY MANAGEMENT USING SEGMENTATION

Submitted by

SATHIYA.R
SHEIK
ABDULLAH.B
SIVARAJ.M
SNEHA.K
SRINIVASAN.B
SUBASHRI. R

III YEAR – IT

2020-2021

Memory Management Using Segmentation

AIM:

To write a LINUX/UNIX C Program for the Implementation of Segmentation.

SOURCE CODE:

```
#include<stdio.h>
#include<conio.h>
struct list
{
int seg;
int base;
int limit;
struct list *next;
} *p;
void insert(struct list *q,int base,int limit,int seg)
{
if(p==NULL)
{
p=malloc(sizeof(Struct list));
p->limit=limit;
p->base=base;
p->seg=seg;
p->next=NULL;
}
else
{
while(q->next!=NULL)
{
Q=q->next;
Printf("yes")
}
q->next=malloc(sizeof(Struct list));
q->next ->limit=limit;
q->next ->base=base;
q->next ->seg=seg;
q->next ->next=NULL;
}}
int find(struct list *q,int seg)
{
while(q->seg!=seg)
```

```
{
q=q->next;
}
return q->limit;
}
int search(struct list *q,int seg)
{
while(q->seg!=seg)
{
q=q->next;
}
return q->base;
}
main()
{
p=NULL;
int seg,offset,limit,base,c,s,physical;
printf("Enter segment table/n");
printf("Enter -1 as segment value for termination\n");
do
{
printf("Enter segment number");
scanf("%d",&seg);
if(seg!=-1)
{
printf("Enter base value:");
scanf("%d",&base);
printf("Enter value for limit:");
scanf("%d",&limit);
insert(p,base,lmit,seg);
}}
while(seg!=-1)
printf("Enter offset:");
scanf("%d",&offset);
printf("Enter bsegmentation number:");
scanf("%d",&seg);
c=find(p,seg);
s=search(p,seg);
if(offset<c)
{
```

IT-P53 DATABASE MANAGEMENT SYSTEMS LABORATORY

Subject Code	Subject Name	Lectures (Periods)	Tutorial (Periods)	Practical (Periods)
IT-P53	Database Management Systems Laboratory	0	0	3

Course Objectives:

To familiarize students with the database systems concepts

1. To design databases for real-time applications
2. To provide students with hands-on experience to understand and to be familiar in Oracle database, SQL, Oracle Reports and Oracle Forms.
3. To understand how to administer a database system
4. To acquire knowledge of JDBC and ODBC connectivity

Course Outcomes:

1. An ability to analyze database needs and functions
2. An ability to create data models
3. An ability to create Entity-Relationship (E-R) diagrams
4. An ability to design and implement databases using database technology
5. An ability to use normalization rules and principles to create normalized databases
6. An ability to manage databases as a DBA

Experiments are to be carried out in DB2 / ORACLE and VB/ Open source DBMS package with the required front end software

1. **Study of Database Concepts**: Relational model – table – operations on tables – index – table space – view – schema – data dictionary – privilege – role – transactions.
2. **Study of SQL**: Primitive Data Types – User Defined data Types – Built-in Functions –To create, alter, drop, select, insert, delete, update, commit, rollback, save point, grant, revoke.
3. **Study of Query Types:** Queries involving Union, Intersection, Difference, Cartesian Product, Divide Operations – Sub-Queries – Join Queries – Nested Queries – Correlated Queries – Recursive Queries. .
4. **Application:** Design and develop any three of the following:
   - Library Information System
   - Logistics Management System
   - Students' Information System
   - Ticket Reservation System
   - Hotel Management System
   - Hospital Management System
   - Inventory Control
   - Retail Shop Management
   - Employee Information System
   - Payroll System
   - Any other Similar System.

Content beyond the Syllabus:

1. Conceptual designing using ERDRAW

Text Books:

1. Oracle developer handbook
2. SQL/PL/SQL for Oracle by P.S. Deshpande IIT Madras,Dream tech Press

Reference Books:

1. Elmasri, Navathe, Fundamentals of database Systems, Sixth Edition, Addison Wesley, 2010.
2. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts",McGraw-Hill International Inc., 2011.

Websites:

1. www.oracle-developer.net
2. www.oracle.com/DBA

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY Puducherry 605 107

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT P53 Data Base Management Systems Lab

1. Installing Oracle Express Edition in Windows OS
2. Creating a new user in Oracle 11g

Submitted by

UMAMAHESWARI.B

B.Tech IT- III Year/ V Semester

(2020-2021)

1. Installing Oracle Express Edition in Windows OS

Aim:

To learn how to install the Oracle Express Edition in their own machines before doing Lab sessions.

Procedure:

Live Demo given to students in Lab session on step by step installation of Oracle Express Edition.

Outcome:

Students installed Oracle 11g in their machines and start working with it..

2. Creating a new user in Oracle 11g

AIM:

To learn how to use the Oracle CREATE USER statement to create a new user in the Oracle database.

Introduction to Oracle CREATE USER statement

The CREATE USER statement allows you to create a new database user which you can use to log in to the Oracle database.

The basic syntax of the CREATE USER statement is as follows:

```
CREATE USER username
  IDENTIFIED BY password
  [DEFAULT TABLESPACE tablespace]
  [QUOTA {size | UNLIMITED} ON tablespace]
  [PROFILE profile]
  [PASSWORD EXPIRE]
  [ACCOUNT {LOCK | UNLOCK}];
```

Code language: SQL (Structured Query Language) (sql)

In this syntax:

CREATE USER username

Specify the name of the user to be created.

IDENTIFIED BY password

Specify a password for the local user to use to log on to the database. Note that you can create an external or global user, which is not covered in this tutorial.

DEFAULT TABLESPACE

Specify the tablespace of the objects such as tables and views that the user will create.

If you skip this clause, the user's objects will be stored in the database default tablespace if available, typically it is USERS tablespace; or the SYSTEM tablespace in case there is no database default tablespace.

QUOTA

Specify the maximum of space in the tablespace that the user can use. You can have multiple QUOTA clauses, each for a tablespace.

Use UNLIMITED if you don't want to restrict the size in the tablespace that user can use.

PROFILE profile

A user profile limits the database resources or password that the user cannot exceed. You can assign a profile to a newly created user. If you skip this clause, Oracle will assign the DEFAULT profile to the user.

PASSWORD EXPIRE

Use the PASSWORD EXPIRE if you want to force the user to change the password for the first time the user logs in to the database.

ACCOUNT {LOCK | UNLOCK}

Use ACCOUNT LOCK if you want to lock user and disable access. On the other hand, specify ACCOUNT UNLOCK to unlock user and enable access.

To execute the CREATE USER statement, you must have the CREATE USER system privilege. Once you create the new user, the privilege domain of the user will be empty. Therefore, if you want to the user to be able to login to the database, you should grant the CREATE SESSION system privilege to the user.

OUTCOME:

Students created more users in their machines and started implementing grant and revoke commands.

IT-T61 COMPUTER NETWORKS

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T61	Computer Networks	3	1	-

Pre-requisite: IT–T35 Digital System and Computer Architecture

Course Objectives: Gets the idea of choosing the required functionality at each layer for a given application and trace the flow of information from one node to another node in the network. Then gives the understanding of division of network functionalities into layers, the component required to build different types of networks and identifying the solution for the functionalities in each layer.

Course Outcomes:

1. Understand the Layered Architecture of Computer Networks.
2. Understand the operation of the main components of computer networks.
3. Learn various network protocols and algorithms.
4. Acquire the required skill to design simple computer networks.

Syllabus:

Unit I

Introduction to Computer Networks: Need for Networking - Service Description –connectionless and Connection-Oriented Services – Circuit and Packet Switching – Access Networks and Physical Media – Wireless Links and Characteristics – Queuing Delay and Packet Loss – Internet Protocol stack – OSI Reference Model - Service Models – History of Computer Networking and the Internet.

Unit II

Application Layer: Principles of Network Applications – The Web and HTTP – FTP – Electronic Mail – SMTP – Mail Message Formats and MIME – DNS – Socket Programming with TCP and UDP. Multimedia Networking: Internet Telephony – RTP – RTCP – RTSP.

Unit III

Transport Layer: Transport Layer Services – Multiplexing and Demultiplexing – UDP – Reliable Data Transfer – Go-Back-N and Selective Repeat. Connection-Oriented Transport: TCP – Segment Structure – RTT estimation – Flow Control – Connection Management – Congestion Control – TCP Delay Modeling – SSL and TLS. Integrated and Differentiated Services: Intserv – Diffserv.

Unit IV

Network Layer: Circuit Switching - Packet Switching Virtual Circuit Switching – IP – ARP – DHCP – ICMP – Routing – RIP – OSPF – Sub netting – CIDR – Inter domain Routing – BGP – IPV6 Basic Features – Inter Domain Multicast – Congestion Avoidance in Network Layer.

Unit V

Data Link Layer– Layer Services– Framing - Error correction and detection – Link Level Flow Control – Medium Access – Ethernet – Token Ring –FDDI – Wireless LAN – Bridges and Switches.

(Total: 60 Periods)

Text Books:

1. James F. Kurose, Keith W. Ross, "Computer Networking, A Top-Down Approach Featuring the Internet", Third Edition, Pearson Education, 2006.
2. Larry L. Peterson, Bruce S. Davie, "Computer Networks: A Systems Approach", Fifth Edition, Morgan Kaufmann Publishers Inc., 2011.
3. William Stallings, "Data and Computer Communications", Eighth Edition, Pearson Education, 2011.

Reference Books:

1. Nader F. Mir, "Computer and Communication Networks", First Edition, PearsonEducation, 2007.
2. Ying-Dar Lin, Ren-Hung Hwang and Fred Baker, Computer Networks: An OpenSource Approach ", McGraw Hill Publisher, 2011.
3. Behrouz A. Forouzan, "Data communication and Networking", Tata McGraw-Hill,2004.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-T61 - Computer Network

CONGESTION CONTROL USING LEAKY BUCKET ALGORITHM

Submitted by

ABDUL RAHIM.S
ABDUL RAHMAN.H
ALTHAF KHAN. J
AMIRTHALAKSHMI.P
ANJANA.V
ARUNA.G

III YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY 605 107

CONGESTION CONTROL USING LEAKY BUCKET ALGORITHM

Aim:

Write a program for congestion control using leaky bucket algorithm

Procedure:

The congesting control algorithms are basically divided into two groups: open loop and closed loop. Open loop solutions attempt to solve the problem by good design, in essence, to make sure it does not occur in the first place. Once the system is up and running, midcourse corrections are not made. Open loop algorithms are further divided into ones that act at source versus ones that act at the destination.

In contrast, closed loop solutions are based on the concept of a feedback loop if there is any congestion. Closed loop algorithms are also divided into two sub categories: explicit feedback and implicit feedback. In explicit feedback algorithms, packets are sent back from the point of congestion to warn the source. In implicit algorithm, the source deduces the existence of congestion by making local observation, such as the time needed for acknowledgment to come back.

The presence of congestion means that the load is (temporarily) greater than the resources (in part of the system) can handle. For subnets that use virtual circuits internally, these methods can be used at the network layer.

Another open loop method to help manage congestion is forcing the packet to be transmitted at a more predictable rate. This approach to congestion management is widely used in ATM networks and is called traffic shaping.

The other method is the leaky bucket algorithm. Each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. In other words, if one or more process are already queued, the new packet is unceremoniously discarded. This arrangement can be built into the hardware interface or simulate d by the host operating system. In fact it is nothing other than a single server queuing system with constant service time.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY ★ PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHAL KUPPAM,
PUDUCHERRY 605 107.

Program:

```
import java.util.*;
public class leakybucketalgorithm {
   public static void main(String[] args)
   {
      Scanner my = new Scanner(System.in);
      int no_groups,bucket_size;
      System.out.print("\n Enter the bucket size : \t");
      bucket_size = my.nextInt();
      System.out.print("\n Enter the no of groups : \t");
      no_groups = my.nextInt();
      int no_packets[] = new int[no_groups];
      int in_bw[] = new int[no_groups];
      int out_bw,reqd_bw=0,tot_packets=0;
      for(int i=0;i<no_groups;i++)
      {
         System.out.print("\n Enter the no of packets for group " + (i+1) + "\t");
         no_packets[i] = my.nextInt();
         System.out.print("\n Enter the input bandwidth for the group " + (i+1) + "\t");
         in_bw[i] = my.nextInt();
         if((tot_packets+no_packets[i])<=bucket_size)
         {
            tot_packets += no_packets[i];
         }
         else
         {
            do
            {
            System.out.println(" Bucket Overflow ");
            System.out.println(" Enter value less than " + (bucket_size-tot_packets));
            no_packets[i] = my.nextInt();
            }while((tot_packets+no_packets[i])>bucket_size);
            tot_packets += no_packets[i];
         }
         reqd_bw += (no_packets[i]*in_bw[i]);
      }
      System.out.println("\nThe total required bandwidth is " + reqd_bw);
      System.out.println("Enter the output bandwidth ");
      out_bw = my.nextInt();
      int temp=reqd_bw;
      int rem_pkts = tot_packets;
      while((out_bw<=temp)&&(rem_pkts>0))
      {
         System.out.println("Data Sent \n" + (--rem_pkts) + " packets remaining");
         System.out.println("Remaining Bandwidth " + (temp -= out_bw));
         if((out_bw>temp)&&(rem_pkts>0))
            System.out.println(rem_pkts + " packet(s) discarded due to insufficient bandwidth");
      }
   }
}
```

IT-T62 WEB TECHNOLOGY

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T62	Web Technology	3	1	0

Course Objectives:

1. To introduce the basics of Network Model.
2. To introduce the Web Development Process and Various Web Technologies.
3. To learn Networking and Security issues of Internet.

Course Outcomes:

On successful completion of this course students will be able to:

1. Use appropriate web development tools for various web application
2. Learn various Networking and Security issues of Internet to have a protected internet use.

UNIT I

Internet principles – Basic Web concepts – Client –Server model – Retrieving data from Internet -Protocols and applications. Web Design process: Web process Model-Goals and problems-design phase-Testing. Site Types and Architecture-Web site types-Dynamic Sites-site structures

UNIT II

Search and Design: Worldwide search-web searching overview-working of search engines-search engine promotion-optimization-Search interface.

Web protocols-HTTPDNS,- Web Servers-components-software, web hosting. Browsers-HTML and scripting languages-cookies- Multimedia in web design.

UNIT III

Web Technologies: Anatomy of xml document - XML markup-working with elements and attributes - creating valid documents-xml objects.

ActiveX controls: Introduction- Building a basic control - OLE and ActiveX- HTML and ActiveX-ActiveX Documents.

UNIT IV

Streaming – Networking Principles – Sockets for Clients - Sockets for Servers – Protocols handlers – Content handlers – Multicast sockets – Remote method invocation.

UNIT V

Internet Security: The Internet-Understanding Firewalls-Hackers-TCP/IP from a security view point –sockets and services-Encryption. Firewall Technology-packet filtering- Network Address Translation-application level proxies-VPN- ideal firewall.

(Total: 60 Periods)

Content beyond Syllabus:

1. Advanced data structures and their implementation.
2. Implementation of the data structures in different language platforms.

Text Books:

1. Thomas A.Powell, The Complete Reference Web design, Tata McGraw-Hill, 2000.
2. Mathew strebe, charlesperkins, Firewalls, BPB, 2000.

Reference Books:

1. Eillotte Rusty Harold, Java Network Programming, O'Reilly Publications, 1997.
2. John paulMueller,Active X from the Ground up, TataMcGraw-Hill,1997.
3. Michael Girdley, Kathryn A. Jones, et al., Web programming with JavaTM, Sams.net publishing, 1996.

Websites:

1. http://www.w3schools.com

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

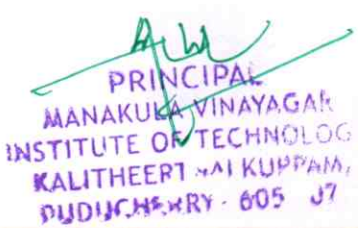
IT – T62 WEB TECHNOLOGY

MINI PROJECT FOR WEB PAGE CREATION

Submitted by

ATHIFFA.M.A
AUROBINDH SAI.C
BALAMURUGAN.P
DEEPIKA.S
DEEPIKA.V
DHIVAGAR.K

III YEAR – IT
(2020-2021)



AIM:

To create a simple webpage using HTML that includes all tags.

A. FRAMES

With frames, you can display more than one HTML document in the same browser window. Each HTML document is called a frame, and each frame is independent of the others.

The Frameset Tag

The <frameset> tag defines how to divide the window into frames.

The Frame Tag

The <frame> tag defines what HTML document to put into each frame.

Example:

```
<frameset cols="25%, 75 %">
<frame src="frame_a.htm">
<frame src="frame_b.htm">
</frameset>
```

Tags and their Description:

<frameset> Defines a set of frames

<frame> Defines a sub window (a frame)

IT-T63 ARTIFICIAL INTELLIGENCE

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T63	Artificial Intelligence	3	1	0

Pre-requisite:
Knowledge in Programming, Discrete mathematics and in probability.

Course Objectives:

1. To search and discover intelligent characteristics of existing AI projects, Intelligent agents map a new problem – as search.
2. To understand different search strategies for a problem.
3. To understand different Knowledge Representation schemes for typical AI problems.
4. To design and implement a typical AI problem to be solved Using Machine Learning Techniques.
5. Implement a futuristic AI application.

Course Outcomes:

On successful completion of this course students will be able to:

1. Capability to develop intelligent systems
2. Apply heuristic concepts to design efficient algorithms that help to attain the goals in satisfactory manner
3. Design applications related to Natural Language Processing and Web applications.

Unit I

Introduction: History of AI - - problem spaces and search- Heuristic Search techniques –Best-first search- Problem reduction- Constraint satisfaction-Means Ends Analysis.
Intelligent agents: Agents and environment – structure of agents and its functions

Unit II

Knowledge Representation: Approaches and issues in knowledge representation- Propositional Logic –Predicate logic-Forward and backward reasoning - Unification- Resolution- Weak slot-filler structure – Strong slot-filler structure- Knowledge- Based Agent

Unit III

Reasoning under uncertainty: Logics of non-monotonic reasoning-Implementation- Basic probability notation - Bayes rule – Certainty factors and rule based systems-Bayesian networks – Dempster - Shafer Theory - Fuzzy Logic.

Unit IV

Planning and Learning: Planning with state space search-partial order planning-planning graphs-conditional planning-continuous planning-Multi-Agent planning. Forms of learning-inductive learning-learning decision trees-ensemble learning-Neural Net learning and Genetic learning

Unit V

Advanced Topics: Game Playing: Minimax search procedure-Adding alpha-beta cutoffs Expert System: Representation-Expert System shells-Knowledge Acquisition. Robotics: Hardware-Robotic Perception-Planning-Application domains

(Total : 60 Periods)

Content beyond Syllabus:

1. Natural language understanding and generation.
2. Speech processing system.

Text Books:

1. Elaine Rich and Kevin Knight and ShivashankarB.Nair, Artificial Intelligence, 3rd edition, Tata Mc Graw Hill, 2009.
2. Ben Coppin, "Artificial Intelligence Illuminated", Jones and Bartlett Publishers, 1st edition, 2004.
3. Stuart J.Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson Education Asia, II edition, 2003.
4. N.P.Padhy, Artificial Intelligence and Intelligent Systems, Oxford University Press, 2nd edition, 2005.

Reference Books:

1. RajendraAkerkar ,Introduction to Artificial Intelligence, Prentice hall of India, 2005.
2. Patrick Henry Winston, Artificial Intelligence, 3rd edition Pearson Education, Inc., 2001.

Websites:

1. http://aima.cs.berkeley.edu/ai.html
2. www.stanford.edu/class/cs221/

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT T63 Artificial Intelligence

Mini Project on validation of Simple Facts using Prolog.

Submitted by

DIVYA.A

ESHWARAMOORTHY.S A

FAYAZ AHAMED.M.S

HARIDHA.S

HARIHARAN.G

HARIHARANE.V

III YEAR – IT

(2020-2021)

AIM: To create simple facts and rules to validate the query using Prolog.

Let's start with a simple example to understand the terminologies. We will provide Facts and Rules to the prolog system and then we will ask queries and we will see what prolog interpreter returns as an answer and why.

INPUT:

Convert the following sentences into facts:

a) burger is a food
b) sandwich is a food
c) pizza is a food
d) sandwich is a lunch
e) pizza is a dinner

Frame the rules for the below sentence:

- Every food is a meal OR Anything is a meal if it is a food

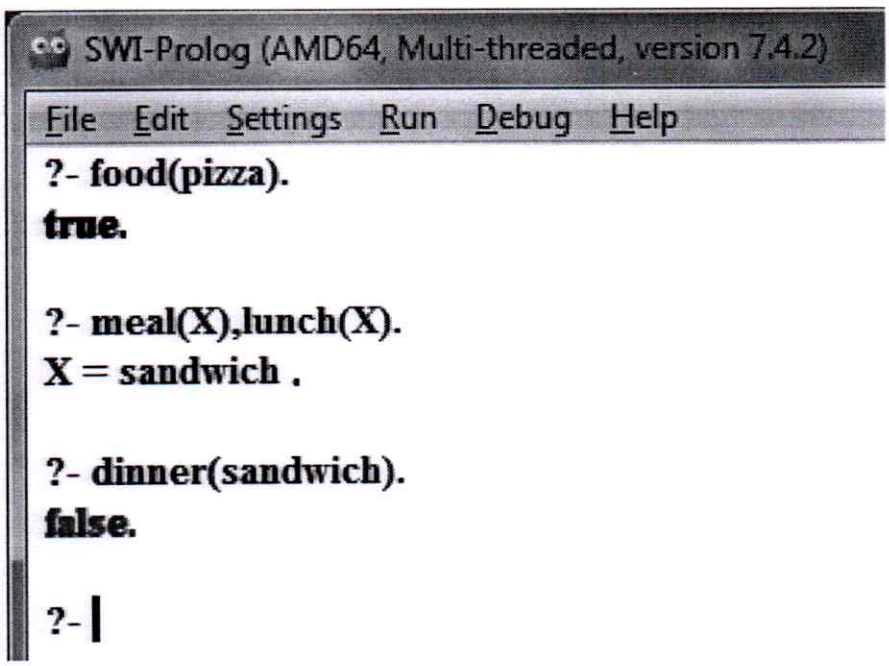
Validate the following query in Query window:

- Is pizza a food?
- Which food is meal and lunch? OR What is both meal and lunch?
- Is sandwich a dinner?

Facts	English meanings of Facts, Rules & Goals
food(burger).	// burger is a food
food(sandwich).	// sandwich is a food
food(pizza).	// pizza is a food
lunch(sandwich).	// sandwich is a lunch
dinner(pizza).	// pizza is a dinner
Rules	
meal(X) :- food(X).	// Every food is a meal OR Anything is a meal if it is a food
Queries / Goals & answers	

?- food(pizza). true.	// Is pizza a food? Answer : true Explanation : Here prolog will return 'true or yes'. Because first, prolog interpreter will trace through the facts and rules in top-down manner and when it can find the match it will provide the answer and in this case it can find the exact match.
?- meal(X), lunch(X). X = sandwich.	// Which food is meal and lunch? OR What is both meal and lunch? Answer : X = sandwich. Explanation : Here in this query we have provided two subgoals where "," comma means 'and'. Prolog always tries to satisfy subgoals in left-to-right manner, so first try to get left most goal i.e. meal(X). But meal(X) rule says - X is a meal if X is a food. So, now we will look for food(X). Here X is a variable and it can bound with any related value. So, in food(X) - X can be burger, sandwich or pizza as per our facts. But second goal says that X should also be lunch. Now we look for X value in lunch(X) and i.e. sandwich. So now we find which food(X) values matches with lunch(X) and the answer is sandwich. You can learn more about these kind of search process
?- dinner(sandwich). false.	// Is sandwich a dinner? Answer : false. Explanation : In this case prolog will find the 'dinner' predicate and will match the argument inside the bracket. But it will return 'false or no' since it cannot find the match.

OUTPUT:



Result: Thus created simple facts and rules to validate the query using Prolog.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

IT-E66 OBJECT ORIENTED ANALYSIS AND DESIGN

Pre-requisite:
Knowledge in the features of Object Oriented Programming Languages

Course Objectives:
To familiarize the students to carry out object oriented analysis and design for developing object oriented software projects

Course Outcomes:
Students acquire the skills to apply Industry recommended Unified Modeling Language Practices for OOAD and document them effectively

Unit I

Object Oriented Methodologies: Software System Life Cycle – Traditional cycle models – Object Oriented approach – Rambaugh et al Object Modeling Technique – Booch Methodology – Jacobson et al methodology –Rational Unified Process (RUP) – Unified Modeling Language (UML) – UML Models.

Unit II

UML Diagrams: Use case diagram – UML class diagram – interaction diagram – state diagram – activity diagram – Requirements for ATM banking system – case study

Unit III

Object Oriented Analysis: Use case driven Object analysis – approaches for identifying classes – identifying objects, relationships attributes, methods for ATM banking system –Object oriented design process – design axioms.

Unit IV

Object Oriented Design: Designing Classes, methods – access layer object storage and object interoperability –access layer for the ATM banking system View layer – designing interface objects – prototyping User interface – view layer for the ATM banking system

Unit V

Design Patterns: Design Patterns – Describing design patterns - catalog of design patterns – organizing the catalog – How design patterns solve design problems – How to select a design pattern – How to use a design pattern – creational pattern : Abstract factory – structural pattern : Adapter – behavioral pattern : chain of responsibility.

(Total : 60 Periods)

Content beyond Syllabus:
1. Students are encouraged to prepare the document for Mini project and Final year project applying OOAD for the system they implement.
2. Using CASE tools for performing OOAD.

Text Books:

1. Ali Bahrami, Object Oriented systems development, Tata Mcgraw Hill Education Private Ltd, 1999.
2. Carol Britton and Jill Doake, A student Gide to Object Oriented Development, Elsevier, Butterworth – Heinemann, Eighth Edition, 2007.
3. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns – elements of reusable object oriented software, Addition Wesley, 1994.

Reference Books:

1. Craig Larman,"Applying UML and Patterns: An Introduction to object-oriented Analysis and Design and iterative development", Third Edition, Pearson Education, 2005
2. Mike O'Docherty "Object-Oriented Analysis & design – understanding system development with UML 2.0", John Wiley, 2005.
3. Grady Booch, James Rumbagh, IvarJacobson, "The UML user Guide", Pearson Education, 2005
4. Timothy C. Lethbridge, Robert Laganiere" Object-Oriented Software Engineering – A practical software development using UML and Java", Tata McGraw-Hill, New Delhi, March 2003.
5. David William Brown, "An Introduction to Object Oriented Analysis Objects and UML in Plain English", 2nd Edition, Wiley, 2001

Websites:
1. www.omg.org
2. http://www.ibm.com/developerworks/rational/products/rose/
3. http://www.smartdraw.com/resources/tutorials/jacobson-oose-

https://slideplayer.com/slide/12505400/

https://fdocuments.in/document/ooad-unit-4-ppt.html

https://www.scribd.com/presentation/384543963/Unit-4-OOAD

https://slideplayer.com/slide/12505400/

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT - E66 OBJECT ORIENTED ANALYSIS AND DESIGN

Mini Project on Hospital Management System

(Sequence Diagram)

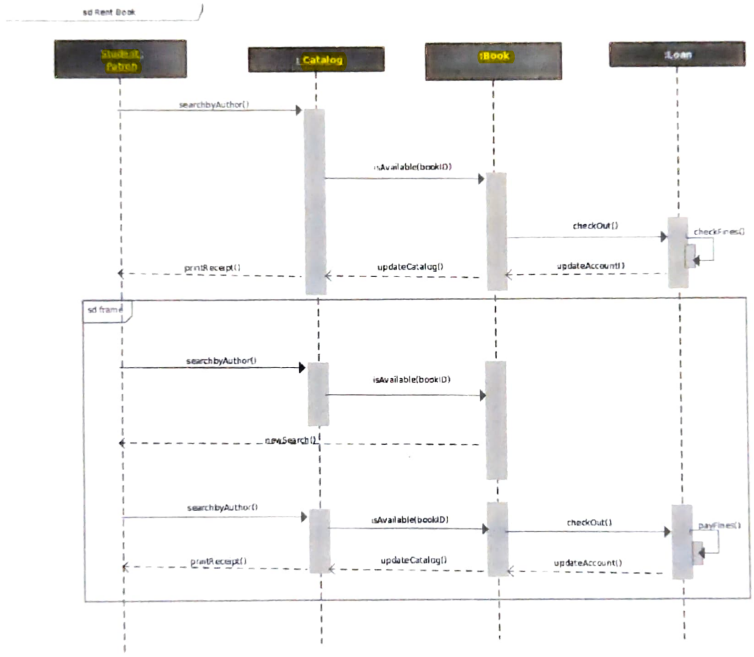
Submitted by

LAVANYA
MAHALAKSHMI.S
MAHALAKSHMI.S
MAHESWARI.S
NAGARAJ
PARTHIBAN.R

III YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM,
PUDUCHERRY 605 107



IT-E68 USER INTERFACE DESIGN

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-E68	User Interface Design	3	1	0

Course Objectives:

1) To study the basic characteristics of graphics and web interfaces, Human Computer Interaction, multimedia interfaces for the web and the principles of evaluating interfaces.

Course Outcomes:

On successful completion of this course students will be able to:

1) The students learn concepts of user interface and used for web applications, human interfaces and for multimedia interfaces.

UNIT I

Introduction: A Taxonomy Of Software Design–Goal-Directed Design– TheThreeModels–VisualInterfaceDesign–Forms–IdiomsAndAffordances–HistoryofRectanglesontheScreen –Windows- Files– Storage and Retrieval Systems – Choosing Platforms.

UNIT II

Interface Design: BehaviorofPresentation–OrchestrationandFlow–TechniquesforInducingandMaintaining Flow–CharacteristicsofGoodUserInterface–PosturesandState–Idiocy–TheSecretWeaponofInterfaceDesign.

UNIT III

Mouse Operations: TheInteraction–MouseOperations–Selection–DirectManipulation–Manipulatinggizmos– Repositioning–Resizing and reshaping– Arrowing– Direct-Manipulation visual feedback – Drag-and-Drop.

UNIT IV

Menu Selection: The Cast– The Meaning of Menus–Menu–Dialog Boxes–DialogBox Etiquette–Toolbars–The Gizmos–Imperative and Selection Gizmos–Entry and Display Gizmos–New Gizmos.

UNIT V

Managing Exceptions & Personalization: EliminatingtheErrorMessages–ManagingExceptions–Undo–Troubles–Redo–SpecialUndo Functions– Installation–Configuration–Personalization.

(Total: 60 Periods)

Text Books:

1. Alan Cooper, The Essentials of User Interface Design,Wiley Dream techIndia (P) Ltd., 2002.
2. Ben Schneiderman, Designing theUserInterface,AddisonWesley,2000.

Reference Books:

1. AlanDix, JanetEFinlay,GregoryD.AbowdandRussellBeale, Human-ComputerInteraction,PrenticeHall,3rd Edition,2003.
2. JacobNielson, Usability Engineering, AcademicPress,1993.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHAL KUPPAM,
PUDUCHERRY 605 107

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-E68 USER INTERFACE DESIGN

Mini Project on Location Sensor Application using MIT App Inventor

Submitted by

PRIYADARSHINI.M

PRIYADHARSHINI.S

RAGHUL.T.P

III YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY 605 107.

Location Sensor Application using MIT App Inventor

AIM: To Design and implement a program for Location Sensor Application using MIT APP Inventor.

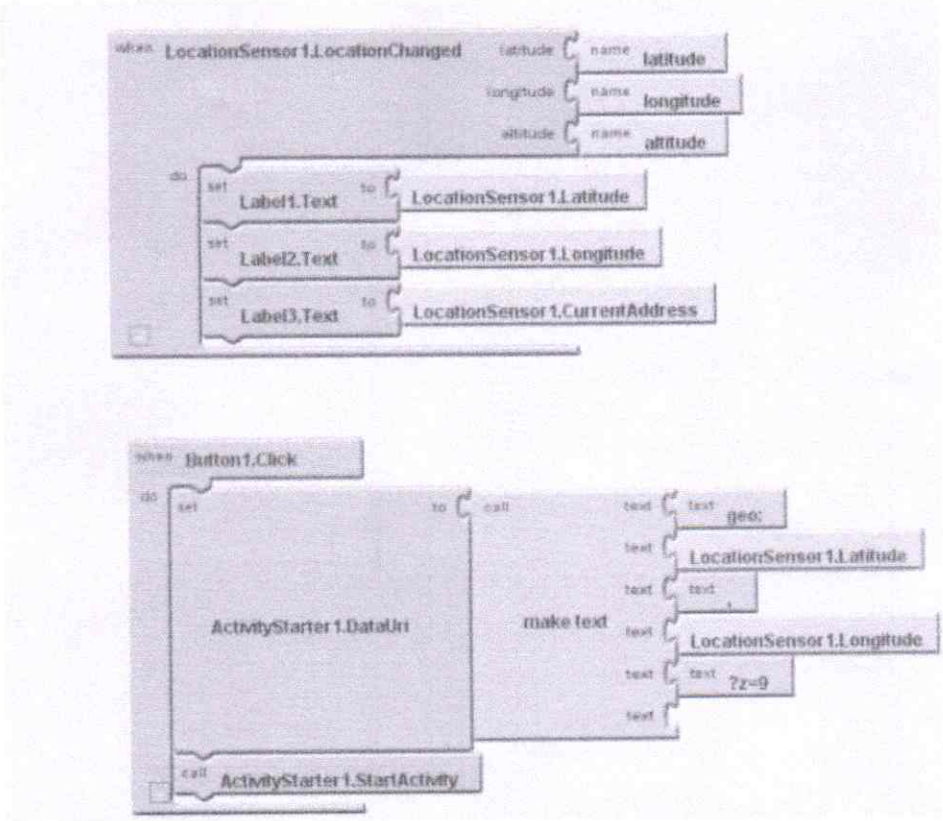
Description / THEORETICAL DISCUSSION:

This application will give the GPS coordinates and (some of the times) address of where you are located in the world. There will be one button and three labels that you will be able to see. And the two hidden components will be LocationSensor and ActivityStarter.

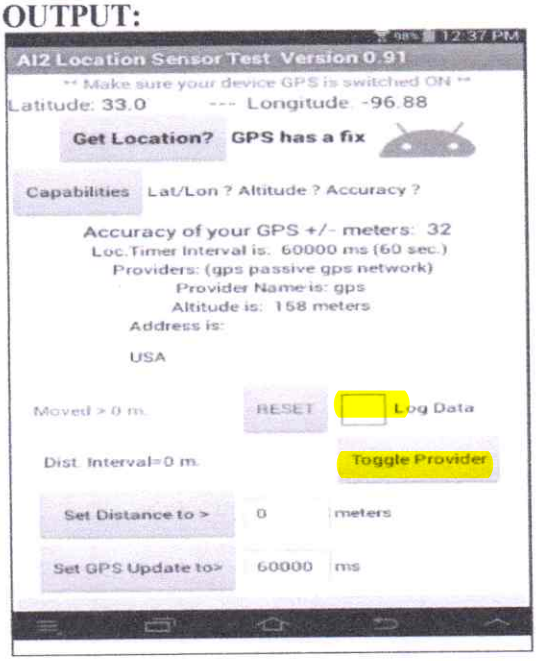
Location sensor: It is a feature that communicates with the GPS of the phone. It can be used in an application by a user to learn lot of things about current location.

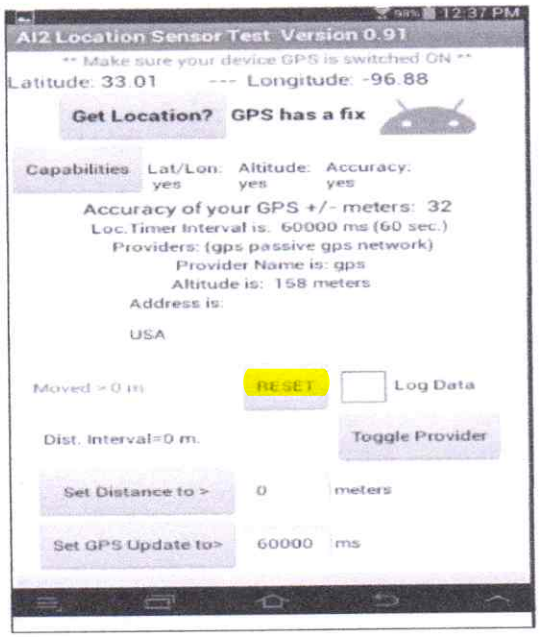
Design Editor Instructions:

1. Drag a button on the screen and name it Button1. This is the button you will press to start Google Maps. On the right side of the screen in the "Components" section, you can see the entity of the button. Click on it, and in the "Properties" section, set the text of the button to be "Find current location".
2. Drag three labels on to the screen, and set their texts to be "Longitude", "Latitude", and "Current Address" respectively.
3. Now go in the "Sensors" section and drag out the "Location sensor" onto the screen. This is an invisible element and will appear at the bottom of the screen.
4. From "other stuff", drag Activity Starter onto the screen. This too is an invisible component and will appear below "Location Sensor".
5. Click on "Location Sensor" and make sure it is enabled. Do the same with "Activity Starter".
6. Name them "LocationSensor1" and "ActivityStarter1" respectively.
7. Location sensor constantly receives updates from the GPS and has information inside it such as current altitude, longitude, latitude, address, accuracy, etc.
8. When you check "enabled" for Location Sensor, it becomes active and can start receiving data from GPS in the background. The way in which this transmission of data happens is part of the location sensor's built-in functionality. We do not need to do anything to facilitate this communication.
9. For the "Activity Starter", in the "Properties" section, set the "Action" field to "android.intent.action. VIEW" Activity Class"field to"com.google.android.maps.MapsActivity and "ActivityPackage" field to "com.google.android.apps.maps"
10. Here, we just specified which activity we want to start and which package the implementation of that activity is located in.



OUTPUT:





OUTCOME:

After Successful Completion of the experiment, we learnt the GPS coordinates address of where we are located in the world.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM,
PUDUCHERRY 605 107

IT-P61 COMPUTER NETWORKS LAB

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P61	Computer Networks Lab	0	0	3

Course Objectives:

1. To learn socket programming
2. To use simulation tools.
3. To analyze the performance of protocols in different layers in computer networks using simulation tools.

1. Applications using TCP Sockets like
   (i) Echo client and echo server
   ii) File transfer
   iii) date and time server & client
   iv) Chat
2. Applications using UDP Sockets like
   ii. DNS
   iii. SNMP
3. Applications using Raw Sockets like
   i) Ping
   ii) Traceroute
4. Programs using RPC
5. Experiments using simulators like OPNET:
   i. Performance comparison of MAC protocols
   ii. Performance comparison of Routing protocols like
   iii. Shortest path routing
   iv. Flooding
   v. Link State
   vi. Hierarchical
   vii. Study of TCP/UDP performance.

(Total: 45 Periods)

Text Books:

1. James F. Kurose, Keith W. Ross, "Computer Networking, A Top-Down Approach Featuring the Internet", Third Edition, Pearson Education, 2006.
2. Larry L. Peterson, Bruce S. Davie, "Computer Networks: A Systems Approach", Fifth Edition, Morgan Kaufmann Publishers Inc., 2011.
3. William Stallings, "Data and Computer Communications", Eighth Edition, Pearson Education, 2011.

Reference Books:

1. Nader F. Mir, "Computer and Communication Networks", First Edition, PearsonEducation, 2007.
2. Ying-Dar Lin, Ren-Hung Hwang and Fred Baker, Computer Networks: An OpenSource Approach ", McGraw Hill Publisher, 2011.
3. Behrouz A. Forouzan, "Data communication and Networking", Tata McGraw-Hill,2004.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT P61 Computer Networks Lab

Mini Project on Simulation of Ring Topology

Submitted by

SANDHIYA.V

SANKARI.S

SANTHIYA.M

SARAN.P

SARANYA.S

SARANYADHEVI.S

III YEAR – IT

(2020-2021)

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

AIM:
To create scenario and study the performance of token ring protocols through simulation.

HARDWARE / SOFTWARE REQUIREMENTS:
NS-2

THEORY:
Star networks are one of the most common computer network topologies. In its simplest form, a star network consists of one central switch, hub or computer, which acts as a conduit to transmit messages. This consists of a central node, to which all other nodes are connected; this central node provides a common connection point for all nodes through a hub. In star topology, every node (computer workstation or any other peripheral) is connected to a central node called a hub or switch. The switch is the server and the peripherals are the clients. Thus, the hub and leaf nodes, and the transmission lines between them, form a graph with the topology of a star. If the central node is passive, the originating node must be able to tolerate the reception of an echo of its own transmission, delayed by the two-way transmission time (i.e. to and from the central node) plus any delay generated in the central node. An active star network has an active central node that usually has the means to prevent echo-related problems.

The star topology reduces the damage caused by line failure by connecting all of the systems to a central node. When applied to a bus-based network, this central hub rebroadcasts all transmissions received from any peripheral node to all peripheral nodes on the network, sometimes including the originating node. All peripheral nodes may thus communicate with all others by transmitting to, and receiving from, the central node only. The failure of a transmission line linking any peripheral node to the central node will result in the isolation of that peripheral node from all others, but the rest of the systems will be unaffected.

ALGORITHM:

1. Create a simulator object
2. Define different colors for different data flows
3. Open a nam trace file and define finish procedure then close the trace file, and execute nam on file.
4. Create six nodes that forms a network numbered from 0 to 5
5. Create duplex links between the nodes to form a STAR Topology
6. Setup TCP Connection between n(1) and n(3)
7. Apply CBR Traffic over TCP
8. Schedule events and run the program.

PROGRAM:

```
#Create a simulator object
set ns [new Simulator]

#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
   global ns nf
   $ns flush-trace
   #Close the trace file
   close $nf
   #Executenam on the trace file
   exec nam out.nam &
   exit0
```

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

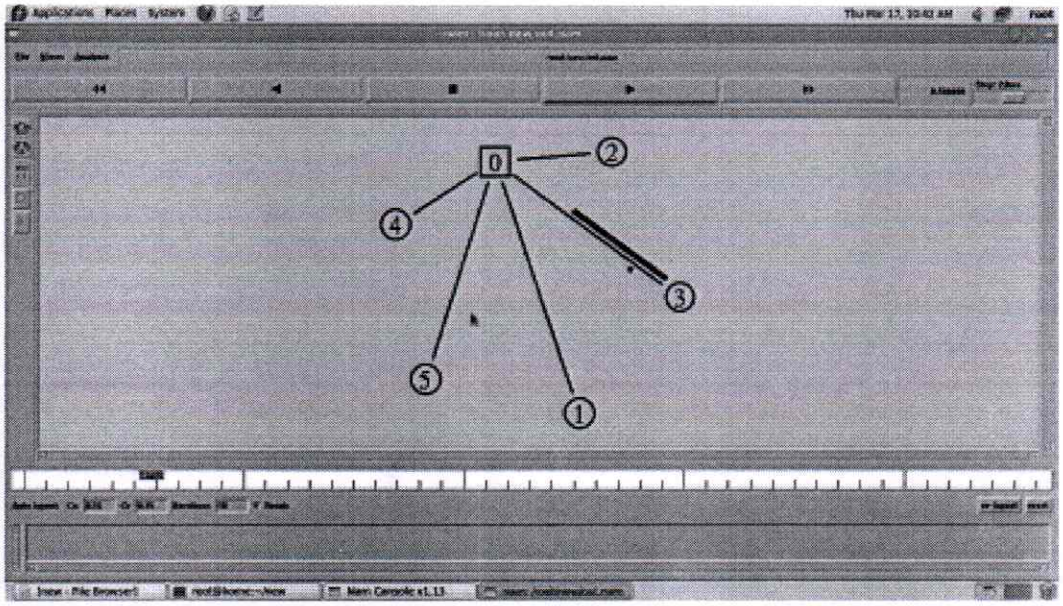
PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

```
}

#Create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

#Change the shape of center node in a star topology
$n0 shape square

#Create links between the nodes
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n0 $n3 1Mb 10ms DropTail
$ns duplex-link $n0 $n4 1Mb 10ms DropTail
$ns duplex-link $n0 $n5 1Mb 10ms DropTail

#Create a TCP agent and attach it to node n0
set tcp0 [new Agent/TCP]
$tcp0 set class_ 1
$ns attach-agent $n1 $tcp0
#Create a TCP Sink agent (a traffic sink) for TCP and attach it to node n3
set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
#Connect the traffic sources with the traffic sink
$ns connect $tcp0 $sink0
# Create a CBR traffic source and attach it to tcp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.01
$cbr0 attach-agent $tcp0
#Schedule events for the CBR agents
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

OUTPUT:



RESULT:

Thus the star Topology was simulated and studied.

IT-P62 WEB TECHNOLOGY LAB

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P62	Web Technology Lab	0	0	3

Course Objectives:

1) To introduce the basics of Network Model.
2) To introduce the Web Development Process and Various Web Technologies.
3) To learn Networking and Security issues of Internet.

Course Outcomes:

On successful completion of this course students will be able to:

1) Use appropriate web development tools for various web application
2) Learn various Networking and Security issues of Internet to have a protected internet use.

Implement the following problems :

1. Creation of HTML Files
2. Working with Client Side Scripting
   - VBScript
   - JavaScript
3. Configuration of web servers
   - Apache
   - Internet Information Server(IIS)
4. Working with ActiveX Controls in web documents.
5. Experiments in JAVA
   - Applets
   - Threads
   - Sockets
6. Working with Server Side Scripting
   - Active Server Pages
   - Java Servelets
7. Sample web application development in the open source environment

Content beyond Syllabus:

1. Advanced data structures and their implementation
2. Implementation of the data structures in different language platforms

Text Books:

1. Thomas A.Powell , The Complete Reference Web design, TataMcGraw-Hill ,2000.
2. Mathew strebe, charlesperkins, Firewalls, BPB, 2000.

Reference Books:

1. Eillotte Rusty Harold, Java Network Programming, O'Reilly Publications, 1997.
2. John paulMueller,Active X from the Ground up, TataMcGraw-Hill,1997.
3. Michael Girdley, Kathryn A. Jones, et al., Web programming with JavaTM, Sams.net publishing, 1996.

Websites:

1. http://www.w3schools.com

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – P62 WEB TECHNOLOGY LAB

ONLINE BOOK STORE SYSTEM

Submitted by

SATHIYA.R
SHEIK ABDULLAH.B
SIVARAJ.M
SNEHA.K
SRINIVASAN.B
SUBASHRI. R

III YEAR – IT

2020-2021

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
KALITHEERTHAL KUPPAM,
PUDUCHERRY - 605 107

ONLINE BOOK STORE SYSTEM

AIM:

Develop static pages (using only HTML) of an online Book store. The pages should resemble: www.amazon.com. The website should consist the following pages.

- Home page
- Registration and user Login
- User profile page
- Books catalog
- Shopping cart
- Payment by credit card Order Conformation

PROCEDURE:

Home page

Main.html:

```
<html>
<head>
<title> Amazon</title>
</head>
<body bgcolor="cyan"> <center>
<strong><h1>Welcome to AMAZON</h1></strong>
<form method="post" action="login.html" target=_blank >
<h4>for books</h4><input type="submit" value="click here">
</form>
</center>
</body>
</html>
```

- Registration and user Login

Login.html:

```
<html>
<head>
<title>
login page</title>
</head>
<body bgcolor="cyan"> <center>
<strong><h1> AMAZON </h1></strong></center>
<right>
<table align="right">

<tr>
<td><h4>user name</td>
<td><input type="text" ></td>
<td></td>
</tr>

<tr>
<td><h4>password</td>
<td><input type="password"></td>
<td></td>
</tr>

<tr>
<td>
<form method="post" action="catalog.html" >
<input type="submit" value="submit" >
</form>
</td>
<td>
<form method="post" action="userpro.html" >
<input type="submit"
value="register" >
  
<input type="reset" value="reset"></form></td>
</tr>

</table>

</body>
</html>
```

IT - P63 MINIPROJECT

Subject Code	Subject Name	I ectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P63	MINI PROJECT	0	0	3

Course Objectives:

Scope of this lab is to understand the application of case tools, which focuses on the following software engineering activities:

- Software requirements analysis and specification
- Software design
- Software implementation
- Software testing and maintenance
- Communication skills and teamwork

Course Outcomes:

On successful completion of this practical's students will be able to:

- Solve any given problem by identifying appropriate Domain/Area
- Prepare SRS for projects
- Prepare SDS for projects
- Document for projects

Exercises:

Students in convenient groups of not more than three members in a group are to take up sample project development activities with the guidelines given below using some of the Computer Aided Software Engineering Tools (CASE):

- Preparing a project – brief proposal including
  - o Problem Identification
  - o Developing a model for solving the problem
  - o A statement of system / process specifications proposed to be developed (Data Flow Diagram)
  - o List of possible solutions including alternatives and constraints
  - o Cost benefit analysis
  - o Time line activities
- A report highlighting the design finalization [based on functional requirements & standards (if any)]
- A presentation including the following
  - o Implementation phase (Hardware / Software / both)
  - o Testing & Validation of the developed system
  - o Learning in the project
- Consolidated report preparation

Content beyond Syllabus:

- Real-Time projects.

Text Books:

1. Theory prescribed books

Reference Books:

1. Theory prescribed Reference

Websites:

1. http://projects.spogel.com/forum_posts.asp?TID=52
2. http://codeincodeblock.blogspot.in/2012/04/list-of-top-ten-college-mini-projects.html

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

ONLINE TOLLGATE PAYMENT SYSTEM

MINI PROJECT LAB

Submitted by

P. SURYA	**REG. NO.: 18TH0651**
K. DHIVAGAR	**REG. NO.: 18TH0613**
T.R. RENGARAJAN	**REG. NO.: 18TH0636**

In partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Of

PONDICHERRY UNIVERSITY

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY - 605 107.

APRIL-2021 (AUGUST 2021)

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that the Mini Project Work titled **"ONLINE TOLLGATE PAYMENT SYSTEM"** is a bonafide work done by **SURYA. P [Reg. No. 18TH0651], DHIVAGAR. K [Reg. No. 18TH0613], RENGARAJAN. T.R [Reg. No. 18TH0636],** in partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology of the Pondicherry University during the academic year 2020-2021.

Staff in charge
(Dr.A.MEIAPPANE)

Head of the Department
(Dr.P.SIVAKUMAR)

Submitted for the University Examination held on 25/08/2021

Internal Examiner

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

External Examiner

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

SOCIAL MEDIA APPLICATION

MINI PROJECT REPORT

Submitted by

V. DEEPIKA	**REG. NO: 18TH0612**
S. SARANYADHEVI	**REG. NO: 18TH0644**
B. UMAMAHESWARI	**REG. NO: 18TH0652**

In partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Of

PONDICHERRY UNIVERSITY

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107.

APRIL- 2021(AUGUST-2021)

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that the Mini Project Work titled **"SOCIAL MEDIA APPLICATION"** is a bonafide work done by **V. DEEPIKA [REG.NO:18TH0612], S. SARANYADHEVI REG.NO: [18TH0644] and B. UMAMAHESWARI [REG.NO:18TH0652]** in partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology of the Pondicherry University during the academic year 2020-21.

Staff in charge
(Dr.A.MEIAPPANE)

Head of the Department
(Dr.P.SIVAKUMAR)

Submitted for the University Examination held on..25.8.21.....

Internal Examiner

External Examiner

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

I

ONLINE MOVIE TICKET RESERVATION
MINI PROJECT REPORT

Submitted by

S.VIGNESH KUMAR	**REG.NO: 18TH0654**
B.SHEIK ABDULLAH	**REG.NO:18TH0646**
M.S.FAYAZ AHAMED	**REG.NO:18TH0616**

In partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Of

PONDICHERRY UNIVERSITY

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

AIRLINES RESERVATION SYSTEM

MINI PROJECT REPORT

Submitted by

HARIDHA.S	REG.NO:18TH0617
LAVANYA.V	REG.NO:18TH0627
YUVASRI.R	REG.NO:18TH0658

In partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

of

PONDICHERRY UNIVERSITY

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107.

APRIL-2021(AUGUST-2021)

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that the Mini Project Work titled"**AIRLINES RESERVATION SYSTEM"** is a bonafide work done by **HARIDHA.S [Reg. No. 18TH0617], LAVANYA.V[Reg. No.18TH0627], and R.YUVASRI [Reg. No. 18TH0658]**in partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology of the Pondicherry University during the academic year 2020-21

23/8/2021

Staff in charge

(Dr.A.MEIAPPANE)

23.8.21.

Head of the Department

(Dr.P.SIVAKUMAR)

Submitted for the University Examination held on..25/08/21.....

25/8/2021

Internal Examiner

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

External Examiner

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY * PUDUCHERRY 605 107.

IT-P71 MOBILE COMPUTING LAB

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P71	Mobile Computing Lab	0	0	3

Course Objectives:
1. To introduce the basics of Mobile computing.
2. To introduce the WML and J2ME Technologies.
3. To learn Bluetooth and distributed mobile computing.

Course Outcomes:
On successful completion of this course students will be able to:
1. Use appropriate mobile communication tools for various mobile application
2. Learn various issues of Mobile Computing

List of Exercises

1. Study of WML and J2ME simulators
2. Design of simple Calculator having +,,,* and / using WML
3. Design of Calendar for any given month and year using WML
4. Design of simple game using WML
5. Animate an image using WML
6. SimulationofapplicationusingJ2MEsimulator
   a. Midlet and other basic UI items.
   b. Bluetooth API
   c. Implementation of Wireless Messaging
   d. MMAPI
7. Simulation of Authentication and encryption technique used in GSM
8. Simulation of applications to access web sites using Microsoft Windows Mobile .net environment.
9. Simulation of Infotainment(news, weather forecasts etc)using WAP
10. Simulation of applications using symbian OS

Course Outcomes:
On successful completion of this course students will be able to:
1. Use appropriate mobile communication tools for various mobile application
2. Learn various issues of Mobile Computing
1. Study of GSM architecture and signalling techniques.
2. Study of Cellular system and related concepts.
3. Study of GPRS services.
4. Study of WAP architecture.
5. Design a web page using WML.
6. Study of Bluetooth architecture.
7. Study of IEEE 802.11 network topology.
8. Study of Distributed mobile computing

Content beyond Syllabus:
1) Advanced cellular systems

Text Books:
1. Reza B Fat and Roy.T. Fielding, "Mobile Computing Principles", Cambridge University Press, 2005.
2. Abdelsalam A Helal, Richard Brice, Bert Haskel, MarekRusinkiewicz, Jeffery L Caster and DarellWoelk, "Anytime, Anywhere Computing, Mobile Computing Concepts and Technology", Springer International Series in Engineering and Computer Science, 2000.

Reference Books:
1. Golden Richard, Frank Adelstein, Sandeep KS Gupta, Golden Richard and Loren Schwiebert, "Fundamentals of Mobile and Pervasive Computing", McGraw-Hill Professional Publishing", 2005.
2. UweHansmann, LotharMerk, Martin S. Nicklons and Thomas Stober, "Principles of Mobile Computing", Springer, 2003.

Websites:
1. http://www.faadooengineers.com/threads/394-MOBILE-COMPUTING-E-bookpresentation-and-lecture-notescovering-full-semester-syllabus
2. http://www.dsc.ufcg.edu.br/~sampaio/cursos/2005.1/BancoDeDados/Artigos/BDMoveis/MobileTransactions/anoverview-of-transaction.pdf

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – P71 MOBILE COMPUTING LAB

Mini Project on

ANDROID APPLICATION DEVELOPMENT

Submitted by

NIVETHA.K

PARTHIBAN .P

PRAVEEN KUMAR.G

RAJNIVASH.M

RAM SUGUMAR. R

SITHARTHAN.S

SIVANESAN. R

IV YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY 605 107

CREATION OF MULTITHREADING IN ANDROID APPLICATION

AIM: To Develop a Mobile application that implements Multi-threading concepts using Android Studio

Algorithm:

1. Using the Android Studio create a project of the type mobile application.
2. Create a simple application in the project.
3. Click on the Design tab and design the prototype of the application.
4. Click on source tab and modify the application logic of the application.
5. Save the project.
6. Right click on the project and click on deploy and undeploy.
7. Then test the android application.

DESCRIPTIONS : When an application is launched, the system creates a thread of execution for the application, called "main." This thread is very important because it is in charge of dispatching events to the appropriate user interface widgets, including drawing events.It is also the thread in which your application interacts with components from the Android UI toolkit (components from the android.widget and android.view packages). As such, the main thread is also sometimes called the UI thread. The system does not create a separate thread for each instance of a component. All components that run in the same process are instantiated in the UI thread, and system calls to each component are dispatched from that thread. When your app performs intensive work in response to user interaction, this single thread model can yield poor performance unless you implement your application properly. Specifically, if everything is happening in the UI thread, performing long operations such as network access or database queries will block the whole UI.When the thread is blocked, no events can be dispatched, including drawing events. From the user's perspective, the application appears to hang. Even worse, if the UI thread is blocked for more than a few seconds (about 5 seconds currently) the user is presented with the infamous "application not responding" (ANR) dialog. Additionally, the Andoid UI toolkit is not thread-safe. So, you must not manipulate your UI from a worker

CREATION OF SIMPLE GPS LOCATION APPLICATION

Aim : To Develop a Mobile application that retrieves the Current GPS location information using Android Studio

Algorithm:

- Using the Android Studio create a project of the type mobile application.
- Create a simple application in the project.
- Click on the Design tab and design the prototype of the application.
- Click on source tab and modify the application logic of the application.
- Save the project.
- Right click on the project and click on deploy and undeploy.
- Then test the android application.

Steps:

For this we will use:

- System location service
- LocationManager and LocationListener
- Permission to access FINE_LOCATION
    - Start a new project "GeoLocation" with an empty activity.
    - Edit the activity_main.xml to change the parent layout to LinearLayout (Verical) and to add the -GPS locating cannot be used indoors.

-If using an emulator you need to pass the location values yourself using the settings of the emulator.

-So, it is better to test it in a device outdoor.

IT-T72 WEB SERVICES AND XML

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T72	Web Services and XML	3	1	---

Pre-requisite: HTML, Component Technology and Databases

Course Objectives:

1. To understand the advantages of using XML technology family.
2. To analyze the problems associated with tightly coupled distributed software architecture.
3. To learn the Web services building block.
4. To implement e-business solutions using XML based web services.

Course Outcomes:

1. Students will understand the benefits of XML, web services and SOA.
2. They will learn how to develop e-business applications using these technologies.

Unit I

XML – benefits – Advantages of XML over HTML, EDI, Databases – XML based standards – Structuring with schemas - DTD – XML Schemas – XML processing – DOM –SAX – presentation technologies – XSL – XFORMS – XHTML – Transformation – XSLT – XLINK – XPATH – XQuery.

Unit II

Roots of SOA – Characteristics of SOA - Comparing SOA to client-server and distributed internet architectures – Anatomy of SOA- How components in an SOA interrelate - Principles of service orientation.

Unit III

Business motivations for web services – B2B – B2C – Technical motivations – limitations of Component Technologies – Architecting web services – Implementation view – web services technology stack – logical view – composition of web services – deployment view – from application server to peer to peer – process view – life in the runtime.

Unit IV

SOA platform basics – SOA support in J2EE – Java API for XML-based web services (JAX-WS) - Java architecture for XML binding (JAXB) – Java API for XML Registries (JAXR) - Java API for XML based RPC (JAX-RPC)- Web Services Interoperability Technologies (WSIT) - SOA support in .NET – Common Language Runtime - ASP.NET web forms – ASP.NET web services – Web Services Enhancements (WSE).

Unit V

WS-BPEL basics – WS-Coordination overview - WS-Choreography, WS-Policy, WS-Security.

(Total : 60 Periods)

Content beyond Syllabus: Semantic web- Xlang- XDBMS

Text Books:

1. AtulKahate," XML and Related technologies", Pearson Education,2008.
2. Thomas Erl, "Service-Oriented Architecture: Concepts, Technology, and Design", Pearson Education, 2005.
3. Newcomer, Lomow, "Understanding SOA with Web Services", Pearson Education, 2005.
4. Ron Schmelzer et al. " XML and Web Services", Pearson Education, 2002.

Reference Books:

1. Keith Ballinger, ". NET Web Services Architecture and Implementation", Pearson Education, 2003.
2. David Chappell, "Understanding .NET A Tutorial and Analysis", Addison Wesley, 2002.
3. KennardScibner and Mark C.Stiver, "Understanding SOAP", SAMS publishing.
4. Sandeep Chatterjee, James Webber, "Developing Enterprise Web Services, An Architect's Guide", Pearson's Edn, 2005.

Websites:

1. http://docs.oracle.com/cd/E17802_01/webservices/webservices/docs/1.6/tutorial/doc/JavaWSTutorial.pdf
2. http://www.w3schools.com/xml/
3. WWW.SOA.COM

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-T72 WEB SERVICE & XML

Program to implement the operation can receive request and will return a Response

Submitted by

DAYANARAJE.S

DEVIKA.M

DHANYA.T

DINESHKUMAR. S

IV YEAR – IT

(2020-2021)

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHAL KUPPAM,
PUDUCHERRY - 605 107.

Program to implement the operation can receive request and will return a Response

Objective

To write a Program to implement the Operations that can Receive Request and will Return a Response.

The **<portType>**element is the most important WSDL element.

WSDL - The <portType> Element

The <portType> element defines **a web service**, the **operations** that can be performed, and the **messages** that are involved.

<portType> defines the connection point to a web service. It can be compared to a function library (or a module, or a class) in a traditional programming language. Each operation can be compared to a function in a traditional programming language.

Operation Types

The request-response type is the most common operation type, but WSDL defines four types:

Type	Definition
One-way	The operation can receive a message but will not return a response
Request-response	The operation can receive a request and will return a response
Solicit-response	The operation can send a request and will wait for a response
Notification	The operation can send a message but will not wait for a response

One-Way Operation

A one-way operation example:

```
<message name="newTermValues">
        <part name="term" type="xs:string"/>
        <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
    <operation name="setTerm">
            <input name="newTerm" message="newTermValues"/>
    </operation>
</portType>
```

In the example above, the portType "glossaryTerms" defines a one-way operation called "setTerm".

The "setTerm" operation allows input of new glossary terms messages using a "newTermValues" message with the input parameters "term" and "value". However, no output is defined for the operation.

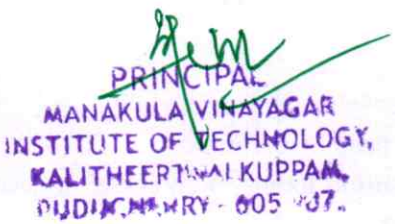
Request-Response Operation

A request-response operation example:

```
<message name="getTermRequest">
        <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
        <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
        <operation name="getTerm">
                <input message="getTermRequest"/>
                <output message="getTermResponse"/>
        </operation>
</portType>
```

In the example above, the portType "glossaryTerms" defines a request-response operation called "getTerm".

The "getTerm" operation requires an input message called "getTermRequest" with a parameter called "term", and will return an output message called "getTermResponse" with a parameter called "value".



IT-T73 CRYPTOGRAPHY AND NETWORK SECURITY

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T73	Information Security	3	1	0

Course Objectives:

To learn about wired and wireless network security with various cryptographic techniques, which include private and public keys algorithms along with attacks types.

Course Outcomes:

On successful completion of this course students will be able to:

1. Use appropriate methods in security
2. Learn various methods of implementing security

UNIT – I CLASSICAL CRYPTOSYSTEM

Security trends – Security Attacks and services – Classical Encryption Techniques — Symmetric cipher model– Basic Number theory –Pseudorandom Number Generation - Stream Ciphers - RC4.

UNIT – II BLOCK CIPHER

Simple DES – DES – Modes of operation – Triple DES – AES – RSA – Attacks – Primality test – factoring.

UNIT – III MESSAGE AUTHENTICATION

Discrete Logarithms – Computing discrete logs – Diffie-Hellman key exchange – ElGamal Public key cryptosystems – Hash functions – Secure Hash - MD5 – Digital signatures – RSA – ElGamal Digital signature scheme.

UNIT – IV NETWORK SECURITY

Key Management and Distribution: X.509, PKI – Electronic Mail security – PGP – IP security – Web Security – SSL, TLS.

UNIT – V WIRELESS NETWORK SECURITY

Wireless Network Security- IEEE 802.11 Wireless LANs - Protocol Overview and Security - Wireless Application Protocol (WAP) - Protocol Overview – Wireless Transport Layer Security (WTLS), WAP end-to-end Security

TOTAL: 60

Content beyond Syllabus:

1. Advanced techniques of security and their implementation
2. Implementation of the latest security for latest security threats

TEXT BOOKS:

1. William Stallings, "Crpyptography and Network security Principles and Practices", Pearson/PHI, 5th ed, 2006. [Unit I, Unit II, Unit IV, Unit V]
2. Wade Trappe, Lawrence C Washington, "Introduction to Cryptography with coding theory", 2nd ed, Pearson, 2007. [Unit III]

REFERENCES:

1. W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education, Second Edition, 2007.
2. Charles P. Pfleeger, Shari Lawrence Pfleeger, "Security in computing", Third Edition – Prentice Hall of India, 2006.
3. Douglas R. Stinson. "Cryptography, theory and practice" , Second edition, CRS Press.

Websites:

1. http://thor.info.uaic.ro/~fltiplea/IS/ICSCourseNotes.html
2. https://www.securityforum.org/
3. eeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=4149673

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

Kalitheerthalkuppam, Madagadipet, Puducherry - 605 107

DEPARTMENT OF INFORMATION TECHNOLOGY

Academic Year 2020- 21- ODD SEM

Subject Name : IT T 73 / Cryptography and Network Security

Mini Project on

Caesar Cipher (shift cipher) substitution technique

Submitted by

HARISH.B
HARSHANAA. U. S
HARSHAVARDENI .D
HASHVANDHINI .K
INSHANIYA BAHADUR .N
JANAVICA. R
JAYABALAGURU. V

IV YEAR – IT

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHAL KUPPAM,
PUDUCHERRY - 605 107

Caesar Cipher (shift cipher) substitution technique

AIM: To implement a program for encrypting a plain text and decrypting a cipher text using Caesar Cipher (shift cipher) substitution technique

Description / THEORETICAL DISCUSSION: The Caesar cipher is one of the earliest known and simplest ciphers. It is a type of substitution cipher in which each letter in the plaintext is 'shifted' a certain number of places down the alphabet. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is named after Julius Caesar, who apparently used it to communicate with his generals. More complex encryption schemes such as the Vigenere employ the Caesar cipher as one element of the encryption process. The widely known ROT13 'encryption' is simply a Caesar cipher with an offset of 13. The Caesar cipher offers essentially no communication security, and it will be shown that it can be easily broken even by hand. To pass an encrypted message from one person to another, it is first necessary that both parties have the 'key' for the cipher, so that the sender may encrypt it and the receiver may decrypt it. For the caesar cipher, the key is the number of characters to shift the cipher alphabet. First we translate all of our characters to numbers, 'a'=0, 'b'=1, 'c'=2, ... , 'z'=25. We can now represent the caesar cipher encryption function, e(x), where x is the character we are encrypting, as:

$C=E(p)= (p+k) \bmod 26$

Where k is the key (the shift) applied to each letter. After applying this function the result is a number which must then be translated back into a letter. The decryption function is :

$P= D(C)=(C-k) \bmod 26$

ALGORITHM:

1. Create and initialize a string ALPHABET that holds the alphabet characters. The index

position of the string represents the numeric representation for the corresponding characters in the string ALPHABET.

2. Read the input plain text to be encrypted and also the Caeser cipher key an integer between 0 and 25.

3. Encrypt the plain text using the Caeser cipher key and the ALPHABET string.

a. For every character in the plain text

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KUPPAM,
PUDUCHERRY 605

i. Search the ALPHABET string for the character and assign the numeric representation of the character (plainnumeric) as the index position of the character in the ALPHABET string.

ii. Perform encryption using

ciphernumeric = (plainnumeric + Caeser cipher key) mod 26

iii. Use ciphernumeric as the index position and get the corresponding character from the ALPHABET string as the equivalent cipher text character for the plain text character

b. Print the equivalent cipher text

4. Decrypt the cipher text using the Caeser cipher key and the ALPHABET string.

a. For every character in the cipher text

i. Search the ALPHABET string for the character and assign the numeric representation of the character (ciphernumeric) as the index position of the character in the ALPHABET string.

ii. Perform decryption using

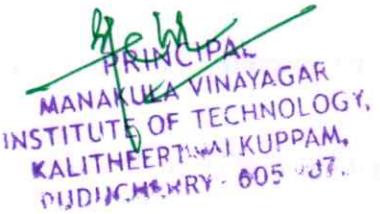
Plainnumeric = (ciphernumeric - Caeser cipher key) mod 26,

if plainnumeric < 0 , plainnumeric = plainnumeric + 26

iii. Use plainnumeric as the index position and get the corresponding character from the ALPHABET string as the equivalent plain text character for the cipher text character

b. Print the equivalent plain text

5. Stop



IT-E72 SOFTWARE TESTING

Subject Code	Subject Name	Lectures (Periods)	Tutorial (Periods)	Practical (Periods)
IT E72	SOFTWARE TESTING	3	1	0

Course Objectives:

1. To learn, practice and apply the software testing industry practices
2. To acquire knowledge on the various test design strategies, levels of testing and test management

Course Outcomes:

1. Ability to apply appropriate testing methods for varying requirements of the software industry
2. Understanding and executing the responsibility of the software testing personal and producing error free software

UNIT I

INTRODUCTION: Testing as an Engineering Activity – Role of Process in Software Quality – Testing as a process – Basic Definitions – Software Testing Principles – The Tester's Role in a Software Development Organization – Origins of Defects – Defect Classes – The Defect Repository and Test Design – Defect Examples – Developer/Tester Support for Developing a Defect Repository.

UNIT II

TEST CASE DESIGN: Introduction to Testing Design Strategies – The Smarter Tester – Test Case Design Strategies – Using Black Box Approach to Test Case Design Random Testing – Requirements based testing – positive and negative testing –– Boundary Value Analysis – decision tables - Equivalence Class Partitioning state-based testing– cause effect graphing – error guessing - compatibility testing – user documentation testing – domain testing Using White–Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing - Coverage and Control Flow Graphs – Covering Code Logic – Paths – Their Role in White–box Based Test Design – code complexity testing – Evaluating Test Adequacy Criteria.

UNIT III

LEVELS OF TESTING: The Need for Levels of Testing – Unit Test – Unit Test Planning –Designing the Unit Tests. The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests – Integration Test Planning – scenario testing – defect bash elimination -System Testing – types of system testing - Acceptance testing – performance testing - Regression Testing – internationalization testing – ad-hoc testing - Alpha – Beta Tests – testing OO systems – usability and accessibility testing

UNIT IV

TEST MANAGEMENT: People and organizational issues in testing – organization structures for testing teams – testing services - Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – test management – test process - Reporting Test Results – The role of three groups in Test Planning and Policy Development – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group.

UNIT V

CONTROLLING AND MONITORING: Software test automation – skills needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation- Test metrics and measurements –project, progress and productivity metrics – Status Meetings – Reports and Control Issues – Criteria for Test Completion – SCM – Types of reviews – Developing a review program – Components of Review Plans– Reporting Review Results. – evaluating software quality – defect prevention – testing maturity model.

(Total: 45 Periods)

Content beyond the Syllabus:

The students can be encouraged to apply concepts learnt in this course in their programming laboratory and project

Text Books:

1) SrinivasanDesikan and Gopalaswamy Ramesh, " Software Testing – Principles and Practices", Pearson education, 2006.
2) AdityaP.Mathur, "Foundations of Software Testing", Pearson Education, 2008.

Reference Books:

1. Boris Beizer, "Software Testing Techniques", Second Edition,Dreamtech, 2003.
2. Elfriede Dustin, "Effective Software Testing", First Edition, Pearson Education, 2003.
3. RenuRajani, Pradeep Oak, "Software Testing – Effective Methods, Tools and Techniques", Tata McGraw Hill, 2004.

Websites:

1. *www.mtsu.edu/~storm*

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – E72 SOFTWARE TESTING

MINI PROJECT ON TO TEST BY SELECTING THE NUMBER OF STUDENTS WHO HAVE SCORED MORE THAN 60 IN ANY ONE SUBJECT (OR ALL SUBJECTS).

Submitted by

KAARTHIK.R
KALAISELVI. R
KALAISRIRAM .S
KANIMOZHI. D
KIRUTHIGAIPRIYA.K
KIRUTHIKA.B
KISHOR KUMAR .J

IV YEAR – IT
(2020-2021)

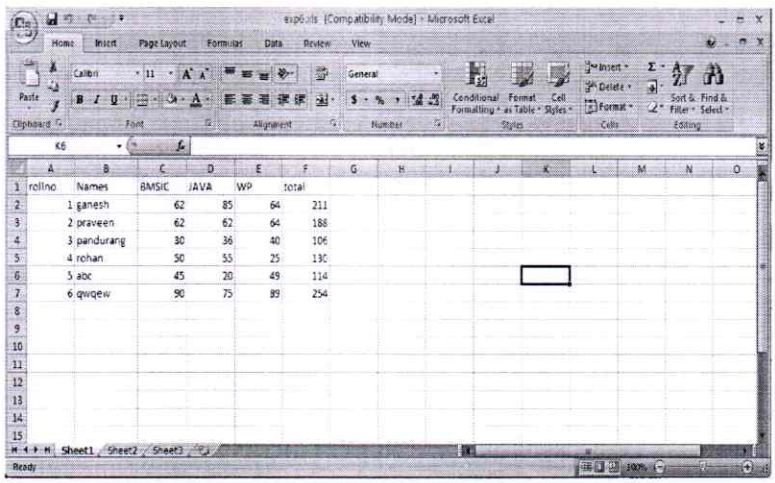
PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).

```
import java.io.FileInputStream; import
java.io.FileOutputStream; import
jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet; import
jxl.write.WritableWorkbook; import
org.testng.annotations.*; public class
exp7 {

@BeforeClass
public void setUp() throws Exception {
}
@Test
public void testImportexport1() throws Exception { FileInputStream fi =
new FileInputStream("D:\\exp6.xls"); Workbook w =
Workbook.getWorkbook(fi);
Sheet s = w.getSheet(0);
String a[][] = new String[s.getRows()][s.getColumns()]; FileOutputStream fo =
new FileOutputStream("D://exp7Result.xls"); WritableWorkbook wwb =
Workbook.createWorkbook(fo); WritableSheet ws = wwb.createSheet("result",
0);
int c=0;
for (int i = 0; i < s.getRows(); i++) { for
(int j = 0; j < s.getColumns(); j++)
{

        if(i >= 1)
        {
                String b= new String();

            b=s.getCell(3,i).getContents(); int
            x= Integer.parseInt(b);
```

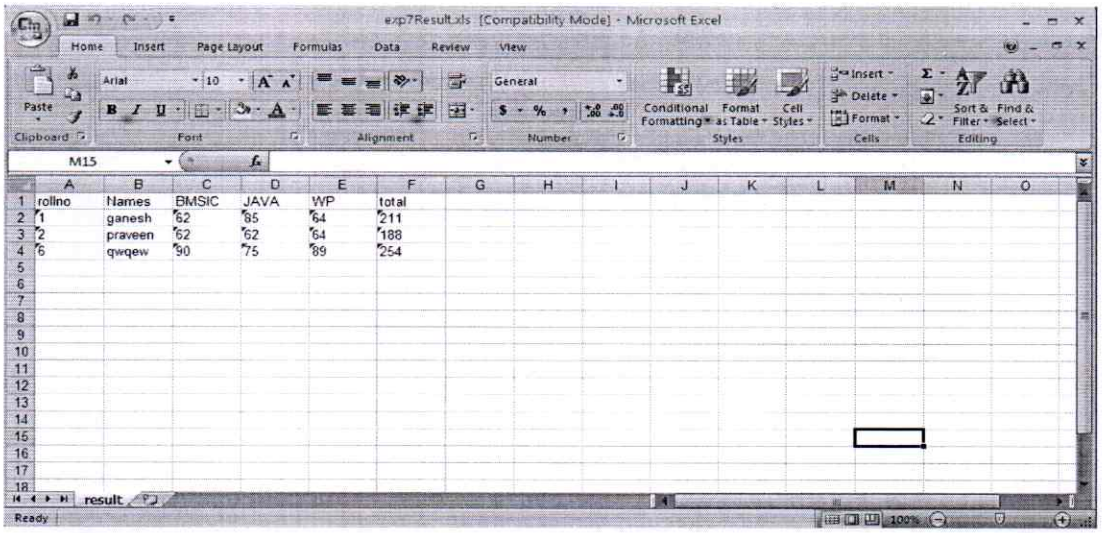
```
        if( x < 60)
        {
                c++;
                break;
        }
        }

                a[i][j] = s.getCell(j, i).getContents();
                        Label l2 = new Label(j, i-c, a[i][j]);
                ws.addCell(l2);

}
}
wwb.write();
wwb.close();
}
}
```

INPUT



	A	B	C	D	E	F
1	rollno	Names	BMSIC	JAVA	WP	total
2	1	ganesh	62	85	64	211
3	2	praveen	62	62	64	188
4	3	pandurang	30	36	40	106
5	4	rohan	50	55	25	130
6	5	abc	45	20	49	114
7	6	qwqew	90	75	89	254

OUTPUT



	A	B	C	D	E	F
1	rollno	Names	BMSIC	JAVA	WP	total
2	1	ganesh	62	85	64	211
3	2	praveen	62	62	64	188
4	6	qwqew	90	75	89	254

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
Kalitheerthalkuppam,
Puducherry - 605 107.

IT-E79 BIG DATABASES

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-E79	Big Databases	3	1	0

Pre-requisite: IT-T54 DBMS

Course Objectives: The students are to understand the concepts of Big Data

Course Outcomes:

1) The students can use the tools of Big Data
2) The students can be able to provide security to Big Data
3) The students can able to turn Big Data into big money

Unit I

Introduction to Big Data: Big Data – The Evolution of Big data - Basics - Big Data Analytics and its Importance – challenges-Issues- Future of Big Data.

Unit II

Basic Big Data Analytic Methods and Modeling: Introduction to "R", analyzing and exploring data with "R"-Modeling: Architecture - Hybrid Data Modeling – Data Computing Modeling.

Unit III

Technology and Tools: MapReduce/Hadoop – NoSQL: Cassandra,HBASE – Apache Mahout – Tools.

Unit IV

Big Data Security: Big Data Security, Compliance, Auditing and Protection: Pragmatic Steps to Securing Big Data, Classifying Data, Protecting Big Data Analytics, Big Data and Compliance, The Intellectual Property Challenge –Big Data in Cyber defense.

Unit V

Case Studies: MapReduce: Simplified Data Processing on Large Clusters- RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Database's - Analytics: The real-world use of big data - New Analysis Practices for Big Data.

(Total: 60 Periods)

Content beyond Syllabus:

To understand the real-time use of Big Data

Text Books:

1. Frank.J.Ohlhorst, "Big Data Analytics : Turning Big Data into Big Money", Wiley &Sas Business Series, 2013

Reference Books:

1. Paul C. Zikopoulos, Chris Eaton, Dirk deRoos, Thomas Deutsch, George Lapis, "Understanding Big Data Analytics for Enterprise Class Hadoop and Streaming Data", The McGraw Hill, 2012.
2. "Planning for Big Data", O'Reilly Radar Team, 2012.
3. "Big Data Now Current Perspectives", O'Reilly Media, 2011.

Websites:

1. http://highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques/
2. http://gigaom.com/2012/12/18/a-programmers-guide-to-big-data-12-tools-to-know/

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY ★ PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – E71 BIG DATABASE

Mini Project on Installation of HADOOP

Submitted by

LAVANYA.D

LOGASOWMYA. V

MADHAN KAVI.M

MADHAVAN.G

MAPPILLAIMEERAN.K

MEENA.S

NANDHINI.M

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
PUDUCHERRY
605 107.

IV YEAR – IT

(2020-2021)

1. Installation of Hadoop:

Hadoop software can be installed in three modes ofoperation:

- **Stand Alone Mode**: Hadoop is a distributed software and is designed to run on a commodity of machines. However, we can install it on a single node in stand-alone mode. In this mode, Hadoop software runs as a single monolithic java process. This mode is extremelyuseful for debugging purpose. You can first testrun your Map-Reduce application in this mode on small data, before actually executing it on cluster with big data.
- **Pseudo Distributed Mode**: In this mode also,Hadoop software is installed on a Single Node.Various daemons of Hadoop will run on the same machine as separate java processes. Hence all the daemons namely NameNode, DataNode, SecondaryNameNode, JobTracker,TaskTracker run on single machine.
- **Fully Distributed Mode**: In Fully Distributed Mode, the daemons NameNode, JobTracker, SecondaryNameNode (Optional and can be run on a separate node) run on the Master Node.The daemons DataNode and TaskTracker runon the Slave Node.

Hadoop Installation: Ubuntu Operating System in stand-alonemode

Steps for Installation

1. sudo apt-get update
2. In this step, we will install latest version of JDK(1.8) on the machine.

The Oracle JDK is the official JDK; however, it is nolonger provided by Oracle as a default installation for Ubuntu. You can still install it using apt-get.

To install any version, first execute the followingcommands:

a. sudo apt-get install python-software-properties

b. sudo add-apt-repository ppa:webupd8team/java

c. sudo apt-get update

Then, depending on the version you want to install,execute one of the following commands:

Oracle JDK 7: `sudo apt-get install oracle-java7-installer`

Oracle JDK 8: `sudo apt-get install oracle-java8-installer`

3. Now, let us setup a new user account for Hadoop

installation. This step is optional, but recommendedbecause it gives you flexibility to have a separate account for Hadoop installation by separating this installation from other software installation

a. sudo adduser hadoop_dev (Upon executing this command, you will prompted to enter the newpassword for this user. Please enter the passwordand enter other details. Don't forget to save the details at the end)

b. su - hadoop_dev(Switches the user fromcurrent user to the new user created i.e Hadoop_dev)

4. Download the latest Hadoop distribution.

a. Visit this URL and choose one of the mirror sites.You can copy the download link and also use "wget" to download it from command prompt:

Wgethttp:// apache.mirrors.lucidnetworks.net/hadoop/

```
common/hadoop-2.7.0/hadoop-2.7.0.tar.gz
```

```
tar xvzf hadoop-2.7.0.tar.gz
```

6. Rename the folder to hadoop2

```
mv hadoop-2.7.0 hadoop2
```

7. Edit configuration file /home/hadoop_dev/ hadoop2/etc/hadoop/hadoop-env.sh and setJAVA_HOME in that file.

a. vim /home/hadoop_dev/hadoop2/etc/hadoop/

```
hadoop-env.sh
```

b. uncomment JAVA_HOME and update it followingline:

export JAVA_HOME=/usr/lib/jvm/java-8- oracle (Please check for your relevant java installation and set this value accordingly. Latestversions of Hadoop require > JDK1.7)

IT-T71 MOBILE COMPUTING

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T71	Mobile Computing	3	1	0

Course Objectives:

1. To teach the basics of mobile computing ideas and best practices.
2. To teach the emerging wireless network standards.
3. To introduce the various models and data management concepts of mobile computing.
4. To learn the routing and secure protocols of mobile networking.

Course Outcomes:

On successful completion of the module students will be able to:

1. Gain basic knowledge in mobile computing.
2. Should have a broader knowledge on 3G.
3. Gain the knowledge on emerging wireless network standards.

Syllabus:

Unit I

INTRODUCTION: Wireless and Mobile Computing Architecture – Limitations of wireless and mobile communication – Wireless Telecommunication Networks: Digital cellular Systems, TDMA - CDMA – Wireless Networking Techniques – Mobility Bandwidth Tradeoffs – Portable Information Appliances.

Unit II

EMERGING WIRELESS NETWORK STANDARDS: 3 G Wireless Networks – State of Industry – Mobility support Software – End User Client Application – Mobility Middleware –Middleware for Application Development - Adaptation and Agents - Service Discovery Middleware – Finding Needed Services - Interoperability and Standardization.

Unit III

MOBILE NETWORKING: Virtual IP Protocols - Loose Source Routing Protocols - Mobile IP – CDPD – GPRS – UMTS Security and Authentication – Quality of Service – Mobile Access to the World Wide Web.

Unit IV

MOBILE DATA MANAGEMENT: Mobile Transactions - Reporting and Co Transactions –Kangaroo Transaction Model – Clustering Model –Isolation only transaction – 2 Tier Transaction Model – Semantic based nomadic transaction processing.

Unit V

MOBILE COMPUTING MODELS: Client Server model – Client/Proxy/Server Model – Disconnected Operation Model – Mobile Agent Model – Thin Client Model – Tools: Java, Brew, Windows CE, WAP, Sybian, and EPOC.

(Total: 60 Periods)

Content beyond the Syllabus:

1. Data dissemination and broadcasting systems.
2. Data synchronization in mobile computing systems.

Text Books:

1. Reza B Fat and Roy.T. Fielding, "Mobile Computing Principles", Cambridge University Press, 2005.
2. Abdelsalam A Helal, Richard Brice, Bert Haskel, MarekRusinkiewicz, Jeffery L Caster and DarellWoelk, "Anytime, Anywhere Computing, Mobile Computing Concepts and Technology", Springer International Series in Engineering and Computer Science, 2000.

Reference Books:

1. Golden Richard, Frank Adelstein, Sandeep KS Gupta, Golden Richard and Loren Schwiebert, "Fundamentals of Mobile and Pervasive Computing", McGraw-Hill Professional Publishing", 2005.
2. UweHansmann, LotharMerk, Martin S. Nicklons and Thomas Stober, "Principles of Mobile Computing", Springer, 2003.

Websites:

1. http://www.faadooengineers.com/threads/394-MOBILE-COMPUTING-E-book presentation-and-lecture-notes-covering-full-semester-syllabus
   http://www.dsc.ufcg.edu.br/~sampaio/cursos/2005.1/BancoDeDados/Artigos/BDMoveis/MobileTransactions/an-overview-of-transaction.pdf

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT T71 MOBILE COMPUTING

Mini Project on

SIMPLE BLUETOOTH CHAT APPLICATION IN ANDROID

Submitted by

GOWRI.D

KOKILAMBAL.S

VANITHA

DHARSHINI.S

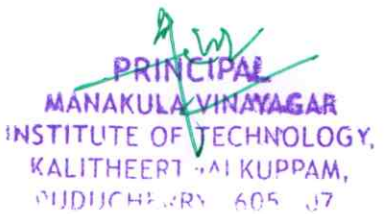
ABINAYA. M

AMUDHA. R

ANUSIYA. J

IV YEAR – IT

(2020-2021)



SIMPLE BLUETOOTH CHAT APPLICATION IN ANDROID

Aim : To create a Simple Bluetooth Chat Application using Android to establish a connection between two are more devices.

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz[4]) from fixed and mobile devices, and building personal area networks (PANs). Invented by telecom vendor Ericsson in 1994,[5] it was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization.

According to this, we can "build" a local are network (LAN) by connecting devices over Bluetooth. The Android platform includes support for the Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices. The application framework provides access to the Bluetooth functionality through the Android Bluetooth APIs. These APIs let applications wirelessly connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features so we absolutely able to transferring data to other devices in the network circle.

Requesting Bluetooth permissions

In order to use Bluetooth service, please add BLUETOOTH permission to your ***AndroidManifest.xml***. Moreover, because we need to discover available devices nearby later, BLUETOOTH_ADMIN permission should be required, too:

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
```

Checking if device supports Bluetooth

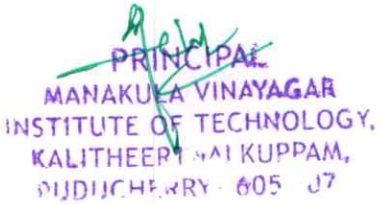
Now to check whether Bluetooth is supported on device or not, we use object of BluetoothAdapter class. If getDefaultAdapter() return null, your device not supports Bluetooth. This is the "check code":

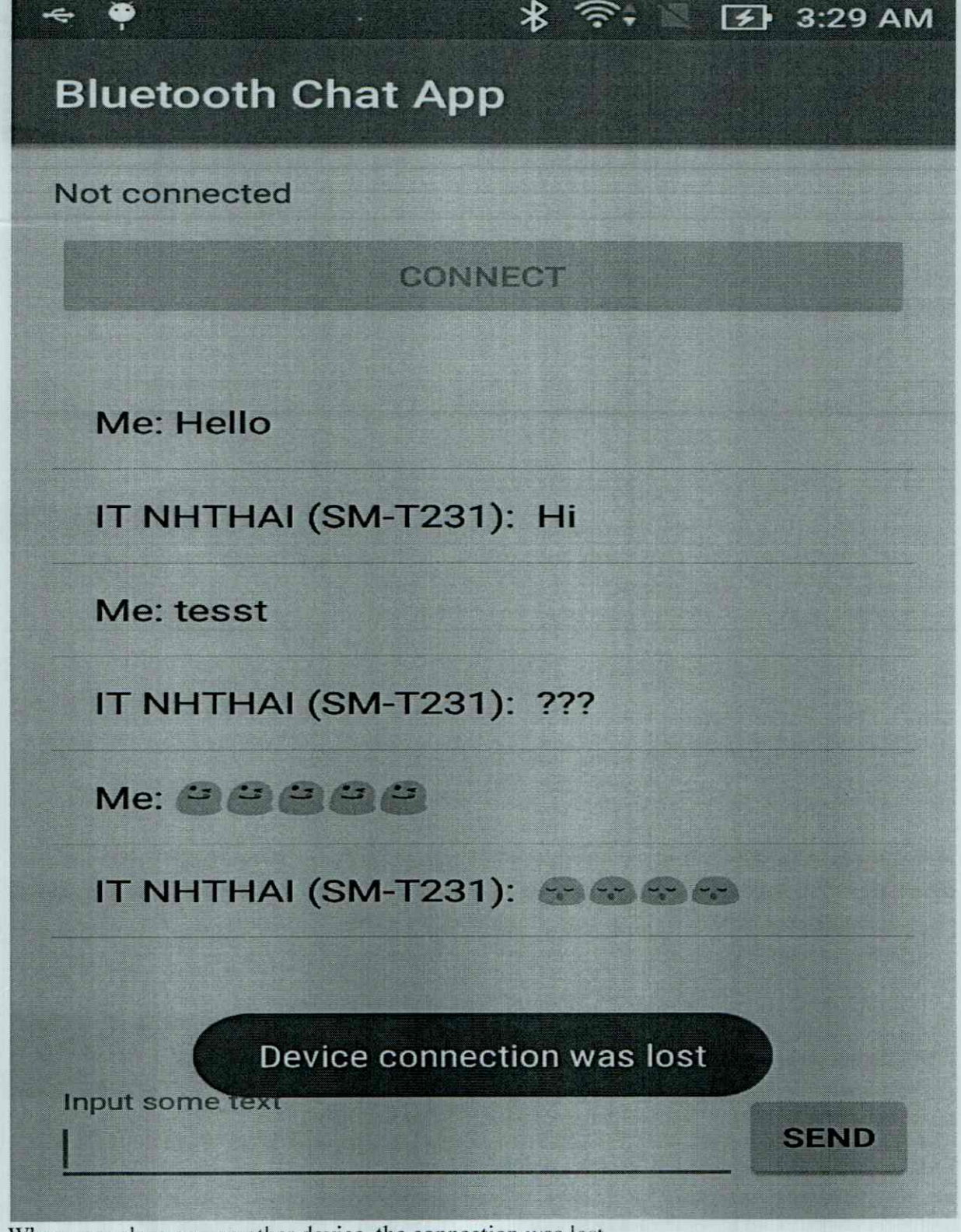
```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    if (bluetoothAdapter == null) {
        Toast.makeText(this, "Bluetooth is not available!", Toast.LENGTH_SHORT).show();
        finish(); //automatic close app if Bluetooth service is not available!
    }
```

Check if Bluetooth is Enabled

The 2nd important works is check if your device is enabled Bluetooth. If not, request to turn it on:

```
if (!bluetoothAdapter.isEnabled())
```





When user close app on other device, the connection was lost

IT-P72 WEB SERVICES AND XML LAB

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P72	Web Services and XML Lab	0	0	3

Course Objectives:

1) The students learn how to design and develop business applications using the popular middleware technologies practiced in the industry.

Course Outcomes:

On successful completion of this course students will be able to:

1) Develop distributed applications in popular platform independent technologies for any business domain.

The students have to develop distributed applications for a given domain using the following technologies:

1. EJB
2. Web Services in Java Platform
3. Web Services with SOA client using C#.net
4. XML with presentation technologies like XSLT, CSS and storage technologies like SAX, DOM with SOAP protocol using C#.net
5. XML with presentation technologies like XSLT, CSS and storage technologies like SAX, DOM with SOAP protocol using Java
6. An interoperable application involving either language/ network protocol heterogeneity or involving any two of the above technologies.

Content beyond Syllabus: Semantic web- Xlang- XDBMS

Text Books:

1. Atul Kahate," XML and Related technologies", Pearson Education,2008.
2. Thomas Erl, "Service-Oriented Architecture: Concepts, Technology, and Design", Pearson Education, 2005.
3. Newcomer, Lomow, "Understanding SOA with Web Services", Pearson Education, 2005.
4. Ron Schmelzer et al. " XML and Web Services", Pearson Education, 2002.

Reference Books:

1. Keith Ballinger, ". NET Web Services Architecture and Implementation", Pearson Education, 2003.
2. David Chappell, "Understanding .NET A Tutorial and Analysis", Addison Wesley, 2002.
3. Kennard Scibner and Mark C.Stiver, "Understanding SOAP", SAMS publishing.
4. Sandeep Chatterjee, James Webber, "Developing Enterprise Web Services, An Architect's Guide", Pearson's Edn, 2005.

Websites:

1. http://docs.oracle.com/cd/E17802_01/webservices/webservices/docs/1.6/tutorial/doc/JavaWSTutorial.pdf
2. http://www.w3schools.com/xml/
3. WWW.SOA.COM

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHAL KUPPAM,
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT - P72 WEB SERVICE & XML LAB

Mini Project on Web Service Creation using JAX-RS

Submitted by

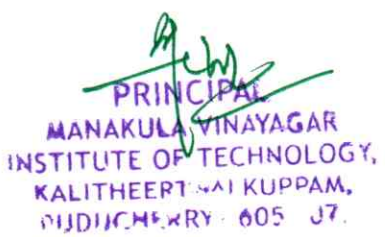
SNEHASHRI.V

SUDHARSANAN. K

SUNDARAVARADHAN.T

IV YEAR – IT

(2020-2021)



Web Service Creation using JAX-RS

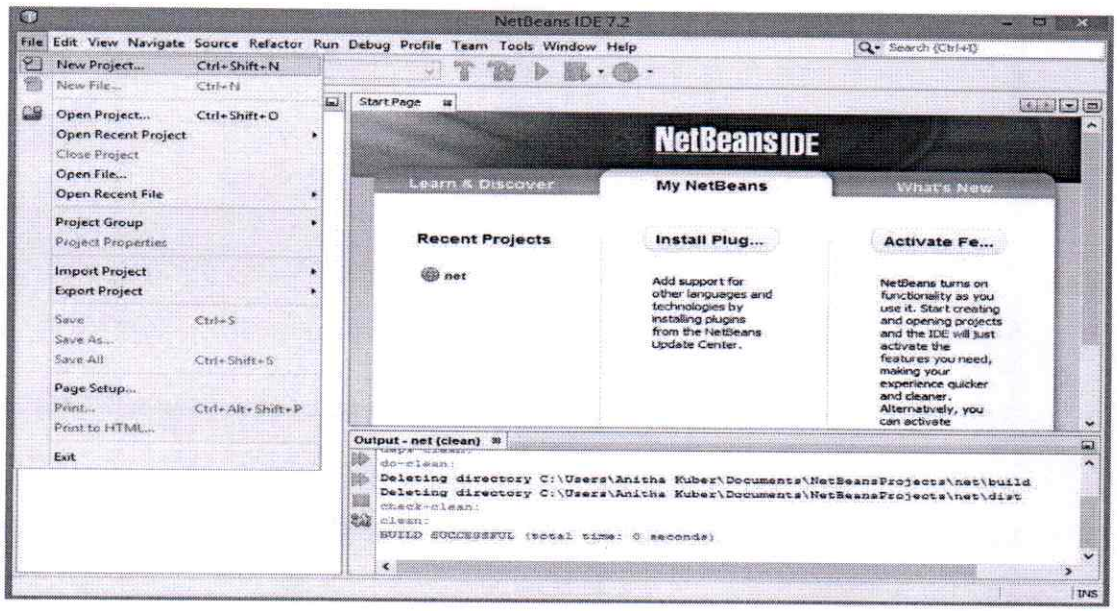
Objective:

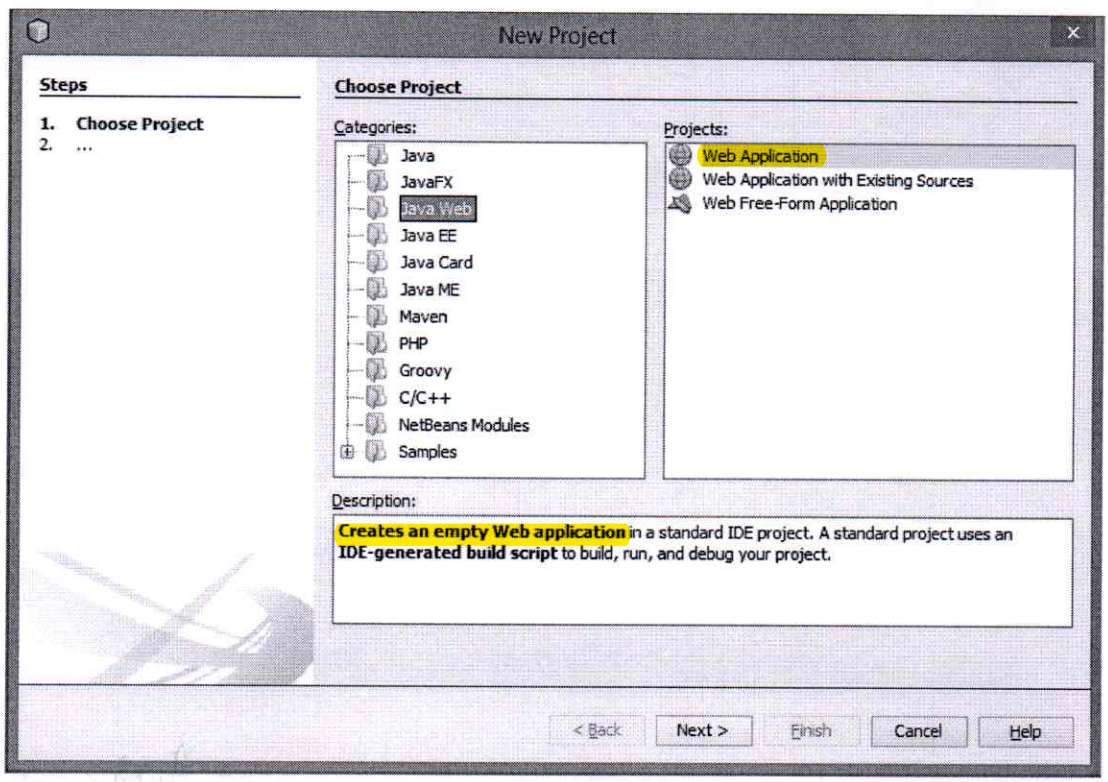
To create a Web Service program using JAX-RS.

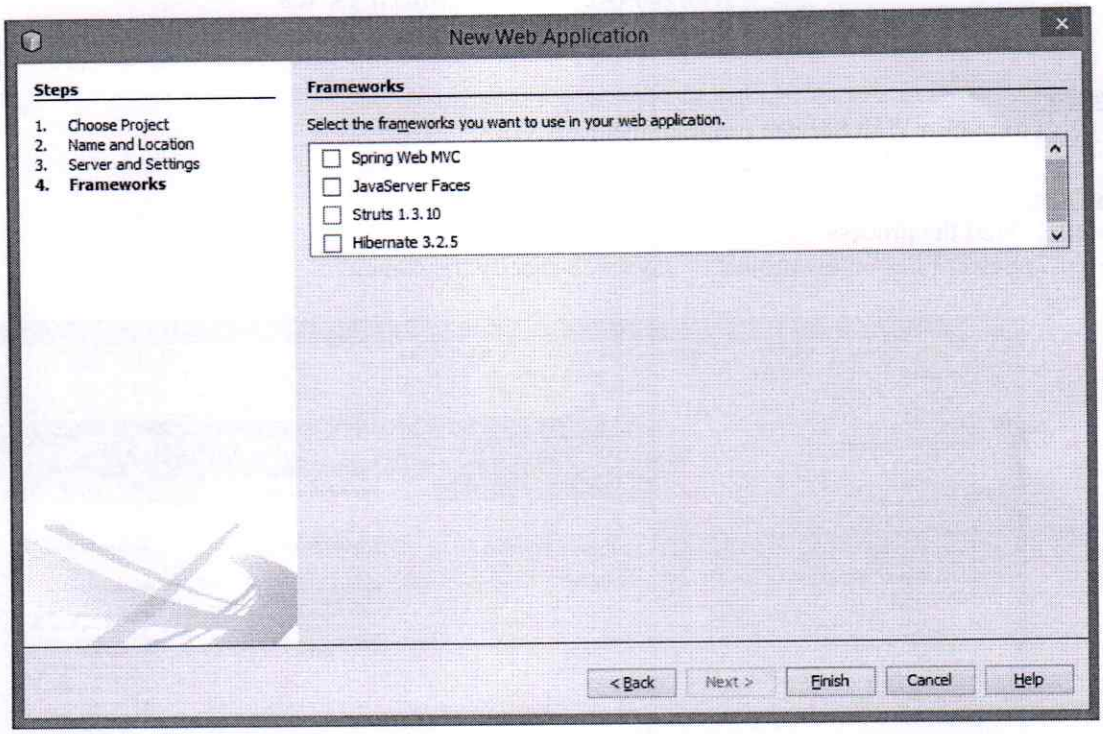
Procedure:

Step-1: Start the process.

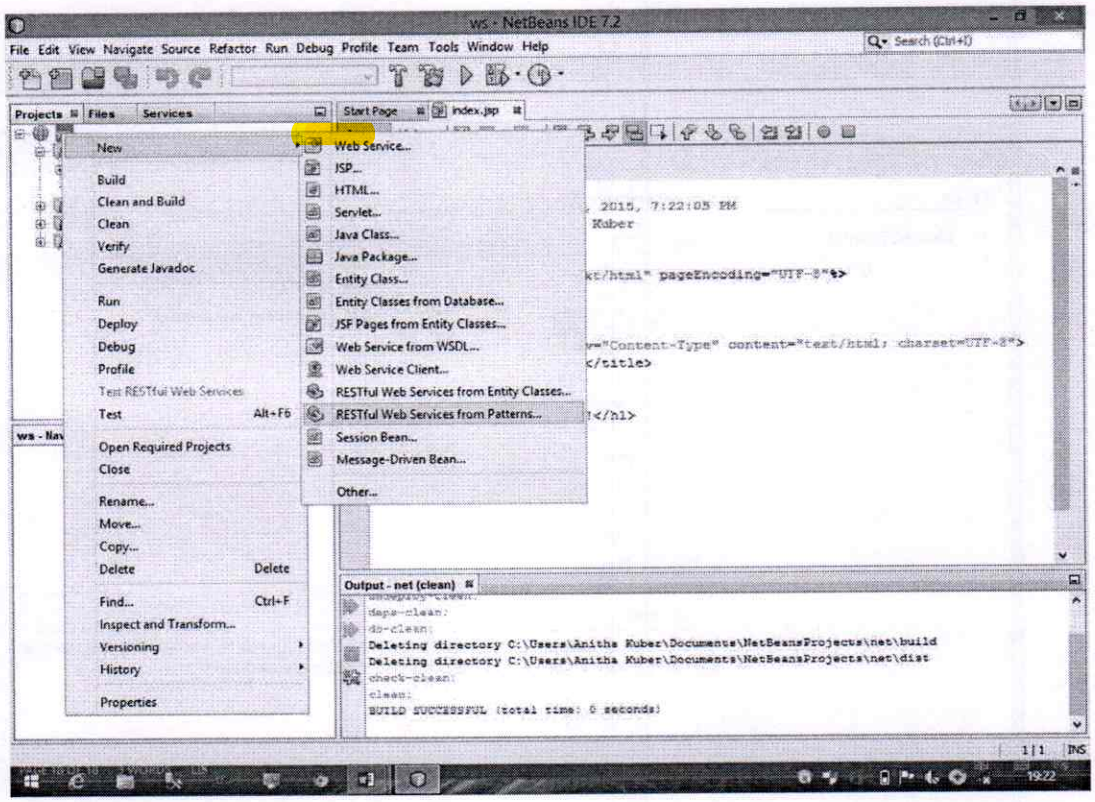
Step-2: Select File-> New project -> Java web specify the name.







Step-3: Right Project->New->RESTFUL web service from pattern.

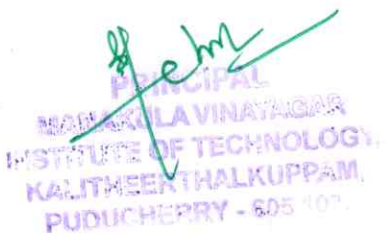


IT-P75INDUSTRIAL TRAINING / INTERNSHIP

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-P75	Industrial Training / Internship	-	-	3

Course Objectives:

In the course of study ,during 5th and6th semesters holidays, each student is expected to undertake a minimum of 4 industrial visits (leading hardware manufacturing /software development companies) and 2 week training or undertake a minimum of one month of industry internship (in a reputed concern). Based on the industrial internships/training/visits, the student has to submit a report at the end of sixth semester highlighting the exposure he/she gained. The report will be evaluated by the departmentalcommitteefor100marks. More weightage will be given for Internship. The proofs for having undergone visits/training are to been closed along with report as enclosures.



Manakula Vinayagar Institute of Technology, Kalitheerthalkuppam, Puducherry – 605 107

Report on Internship

Name of the Student	DIVYABHARATHI. J
Register Number	17TH1209
Year / Department / Section	IV / IT
Duration	3 Months
Name of the Organization	Green Point Global
Objective	To have an knowledge about various web development front end tools.
Tasks and Responsibilities	I was assigned to learn about various web development front end tools.
Courses relevant to work carried out during Internship	web servie, web Technology

Brief report about Internship:

I got an opportunity to do internship in the domain of web development where gained hands on experience in developing online application.

Benefit and Knowledge Gained:

Acquired knowledge about sublime text chrome developer tools and Angular.

Divyabharathi

Date: 5.7.2021

Signature of the Student

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

GreenPoint> Global

Document Ref No: HRES-HR-2027IN V1.0

30th June 2021

Experience Certificate

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Ms. **Divyabharathi J** has successfully completed her internship as **"Intern-Software Developer"** in our organization from 7th April 2021 till 30th June 2021.

During this period, Divyabharathi' s conduct has always been good. Her monthly stipend at the time of her internship was **Rs. 10,000/-** per month (Ten Thousand) only.

We wish her success in all her future endeavours.

For Greenpoint Technology Services (I) Pvt Ltd

Subrata Rath

General Manager – Human Resources

P.S: This is a digitally signed system-generated document, it does not require a physical signature.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHAL KUPPAM,
PUDUCHERRY - 605 107.

IT-T81 PROFESSIONAL ETHICS

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T81	Professional Ethics	3	0	0

Course Objectives:

1. To introduce the basics of Moral Ethics, Engineering Ethics.
2. To introduce the professional Ethics and Case Studies

Course Outcomes:

On successful completion of this course students will be able to:

1. Understand the Values of Moral, Engineering and Professional Ethics

The course should cover the following topics by way of Seminars, Expert Lectures and Assignments:

1. Engineering Ethics – Moral issues, Ethical theories and their uses
2. Engineering as Experimentation – Code of Ethics
3. Engineer's responsibility for safety
4. Responsibilities and rights
5. Global issues of engineering ethics

Content beyond Syllabus:

Case studies on Moral, Engineering and Professional Ethics

Text Books:

1. Charles D.Fleddermann, Engineering Ethics, Prentice Hall, New Mexico, 1999.

Reference Books:

1. Mike W. Martin, Roland Schinzinger, Ethics in Engineering, Tata McGraw Hill, New Delhi, 2005.

Puducherry 605 107

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

Kalitheerthalkuppam, Madagadipet, Puducherry - 605 107

DEPARTMENT OF INFORMATION TECHNOLOGY

IT-T81 PROFESSIONAL ETHICS

Professional Ethics seminar on "Types of Enquiry and Moral Dilemmas"

Submitted by

MADHAN KAVI.M [17TH1231]

MAPPILLAIMEERAN.K [17TH1233]

MEENA.S [17TH1234]

NANDHINI.M [17TH1235]

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHAL KUPPAM, PUDUCHERRY-605107

DEPARTMENT OF INFORMATION TECHNOLOGY

PROFESSIONAL ETHICS: TYPES OF INQUIRY AND MORAL DILEMMAS

INQUIRY

Inquiry means an investigation- Enginerring ethics involves investigations into values,meaning and facts

MORAL DILEMMAS

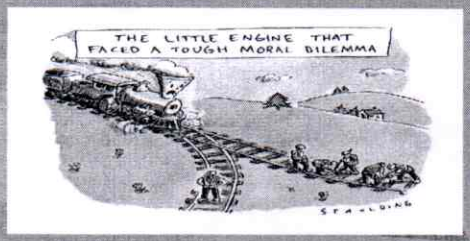
At times, the situations occur where one cannot make immediate decisions as the moral reasons come into conflict. The moral reasons can be rights, duties, goods or obligations, which make the decision making complex.

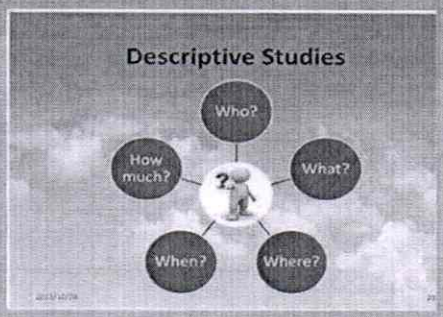
TYPES OF INQUIRY

- ❖Normative Inquiries
- ❖Conceptual Inquiries
- ❖Factual or Descriptive inquiries.

TYPES OF COMPLEXITIES

- ✓Vagueness
- ✓Conflicting reasons
- ✓ Disagreement





STEPS IN FACING MORAL DILEMMAS

- ➢Identification
- ➢Ranking
- ➢Inquiries
- ➢Discussions

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY Puducherry 605 107

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

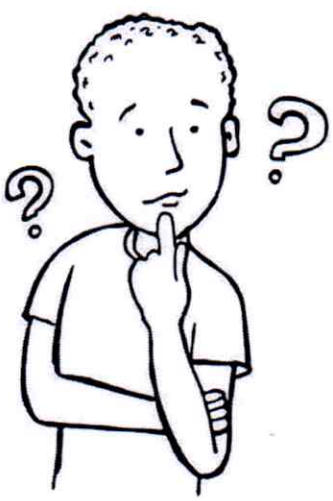
INQUIRY

- Inquiry means an investigation-Enginerring ethics involves investigations into values, meaning and facts.

Types of Inquiry:

- Normative Inquiries
- Conceptual Inquiries
- Factual or Descriptive inquiries.

MORAL DILEMMA



At times, the situations occur where one cannot make immediate decisions as the moral reasons come into conflict. The moral reasons can be rights, duties, goods or obligations, which make the decision making complex.

IT-E83DATA MINING

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-E83	Data Mining	3	1	0

Course Objectives: This course has been designed

1. To introduce the concept of data mining with in detail coverage of basic tasks, metrics, issues, and implication. Core topics like classification, clustering and association rules are exhaustively dealt with.
2. To introduce the concept of data warehousing with special emphasis on architecture and design.

Course Outcomes: On successful completion of this course students will be able to deals with evolving multidimensional intelligent model from a typical system, representation of multi dimensional data for a data warehouse, discovering the knowledge imbibed in the high dimensional system, finding the hidden interesting patterns in data, and gives the idea to evaluate various mining techniques on complex data objects.

Unit I

Introduction: Definition of data mining - data mining vs query tools – machine learning – taxonomy of data mining tasks – steps in data mining process – overview of data mining techniques.

Unit II

Data Warehousing: Definition – Multidimensional Data Model – Data Cube – Dimension Modelling– OLAP Operations – Warehouse Schema – Data Warehouse Architecture – Data Mart – Meta Data – Types of Meta Data – Data Warehouse Backend Process – Development Life Cycle.

Unit III

Data Pre-Processing And Characterization: Data Cleaning – Data Integration and Transformation – Data Reduction – Discretization and Concept Hierarchy Generation – Primitives – Data Mining Query Language – Generalization – Summarization – Analytical Characterization and Comparison - Association Rule – Mining Multi Dimensional data from Transactional Database and Relational Database.

Unit IV

Classification: Classification – Decision Tree Induction – Bayesian Classification – Prediction – Back Propagation – Cluster Analysis – Hierarchical Method – Density Based Method – Grid Based Method – Outlier Analysis.

Unit V

Cluster analysis: Types of data – Clustering Methods – Partitioning methods – Model based clustering methods – outlier analysis. Advanced topics: Web Mining – Web Content Mining – Structure and Usage Mining – Spatial Mining – Time Series and Sequence Mining – Graph Mining

Applications: Case studies in Data Mining applications

(Total : 60 Periods)

Content beyond Syllabus:

1. Handling large data with Grid Computing
2. Creating cloud to handle terabytes of data
3. Hands on demo with recent tools

Text Books:

1. PaulrajPonnaiah, Data Warehousing Fundamentals, Wiley Publishers, Reprint 2011.
2. Jiawei Han, MichelineKamber, Data Mining: Concepts and Techniques, Morgan Kaufman Publishers, 2009.

Reference Books:

1. UsamaM.Fayyad, Gregory Piatetsky Shapiro, Padhrai Smyth, RamasamyUthurusamy, Advances in Knowledge Discover and Data Mining, The M.I.T. Press, 2007.
2. Ralph Kimball, Margy Ross, The Data Warehouse Toolkit, John Wiley and Sons Inc., 2002.
3. Alex Berson, Stephen Smith, Kurt Thearling, Building Data Mining Applications for CRM, Tata McGraw Hill, 2000.
4. Margaret Dunham, Data Mining: Introductory and Advanced Topics, Prentice Hall, 2002.
6. Daniel T. Larose John Wiley & Sons, Hoboken, Discovering Knowledge in Data: An Introduction to Data Mining, New Jersey, 2004.
7. Hand, Mannila and Smyth, Principles of Data Mining, Prentice Hall of India, New Delhi, 2004.
8. Dunham , Data Mining- Introductory and Advanced Topics, Pearson Education, New Delhi, 2003.
9. Arun K Pujari, Data Mining Techniques, Universities press India Pvt Ltd, New Delhi, 2002.
10. Trevor Hastie, Robert Tibshirani, Jerome Friedma, The Elements of Statistical Learning: Data Mining, Inference and Prediction, Prentice Hall, New Delhi, 2002.

Websites:

1. http://dssresources.com/papers/features/langseth/langseth02082004.html
2. http://www-01.ibm.com/software/data/infosphere/data-warehousing/

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY

IT – E83 DATAMINING

Mini Project on APRIORI ALGORITHM

(ASSOCIATION RULE & K-MEANS CLUSTERING)

Submitted by

KIRUTHIKA.B	MADHAVAN.G	PARTHIBAN .P
KISHOR KUMAR .J	MAPPILLAIMEERAN.K	PRAVEEN KUMAR.G
LAVANYA.D	MEENA.S	RAJNIVASH.M
LOGASOWMYA. V	NANDHINI.M	RAM SUGUMAR. R
MADHAN KAVI.M	NIVETHA.K	SITHARTHAN.S

IV YEAR – IT

(2020-2021)

APRIORI ALGORITHM – ASSOCIATION RULE

AIM: To illustrate some of the basic elements of association rule mining using WEKA and The sample dataset used for this example is apriori.arff.

INTRODUCTION

Developed by Agrawal and Srikant 1994

Innovative way to find association rules on large scale, allowing implication outcomes that consist of more than one item

Based on minimum support threshold

Three versions:

1. Apriori (basic version) faster in first iterations
2. AprioriTid faster in later iteratons
3. AprioriHybrid can change from Apriori to AprioriTid after first iterations

LIMITATIONS OF APRIORI ALGORITHM

Needs several iterations of the data

Uses a minimum support threshold

Difficulties to find rarely occuring events

Alternative methods (other than appriori) can address this by using a minimum support thresold

Some competing alternative approaches focus on partition and sampling.

PHASES OF KNOWLEDGE DISCOVERY

Data selection

Data cleansing

Data enrichment (integration with additional resources)

Data transformation or encoding

Data mining

Reporting and display (visualization) of the discovered knowledge

APPLICATION OF DATA MINING

• Data mining can typically be used with transactional databases (for ex. in shopping cart analysis)

• Aim can be to build association rules about the shopping events

• Based on **item sets**, such as {milk, cocoa powder} 2-itemset , {milk, corn flakes, bread} 3-itemset

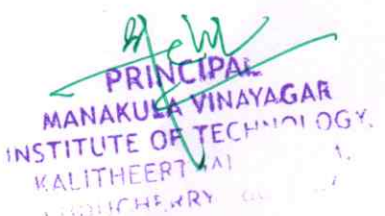
ASSOCIATION RULES

• Items that occur often together can be associated to each other

• These together occuring items form a **frequent itemset**

• Conclusions based on the frequent itemsets form **association rules**

• For ex. {milk, cocoa powder} can bring a rule *cocoa powderèmilk*



K- MEANS CLUSTERING

AIM:

This experiment illustrates the use of simple k-mean clustering with Weka explorer. The sample data set used for this example is based on the vote.arffdata set. This document assumes that appropriate pre-processing has been performed.

WHAT IS CLUSTERING?

- Organizing data into classes such that there is

  high intra-class similarity

  low inter-class similarity

- Finding the class labels and the number of classes directly from the data (in contrast to classification).

- More informally, finding natural groupings among objects.

K-MEANS CLUSTERING

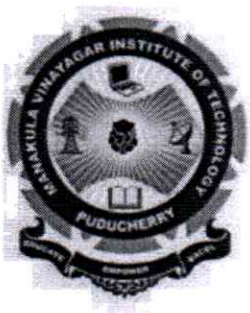
K-Means is simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done.

- ✓ The **k-means algorithm** is an algorithm to cluster *n* objects based on attributes into *k*partitions, where $k<n$.
- ✓ It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.
- ✓ It assumes that the object attributes form a vector space.
- ✓ Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- ✓ K is positive integer number.
- ✓ The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY

DEPARTMENT OF INFORMATION TECHNOLOGY



IT – E83 DATAMINING

Mini Project on

(BAYESIAN CLASSIFICATION & DECISION TREE)

Submitted by

SIVANESAN. R	SWARAJBAL	YUVARAJA .P
SNEHASHRI.V	SWATHI .B	
SUDHARSANAN. K	SWETHA. A	
SUNDARAVARADHAN.T	UMA SANKARI .S	
SURESH. C	YOGANANDHAN.M	

IV YEAR – IT

(2020-2021)

BAYESIAN CLASSIFICATION

AIM:

This experiment illustrates the use of Bayesian classifier with Weka explorer. The sample data set used for this example is based on the weather.nominal.arff data set. This document assumes that appropriate pre-processing has been performed.

BAYESIAN CLASSIFICATION

Bayesian classification is based on Bayes theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class

Bayesian Classification: Why?

A statistical classifier: performs *probabilistic prediction, i.e.,* predicts class membership probabilities

Foundation: Based on Bayes' Theorem.

Performance: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers

Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data.

Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

PROCEDURE:

1. Open the data file in Weka Explorer. It is presumed that the required data fields have been discretized.
2. Next we select the "classify" tab and click choose button to select the "NavieBayes" in the classifier.
3. Now we specify the various parameters. These can be specified by clicking in the text box to the right of the chose button. In this example, we accept the default values his default version does perform some pruning but does not perform error pruning.
4. We select the 10-fold cross validation as our evaluation approach. Since we don't have separate evaluation data set, this is necessary to get a reasonable idea of accuracy of generated model.

5. We now click start to generate the model .the ASCII version of the tree as well as evaluation statistic will appear in the right panel when the model construction is complete.

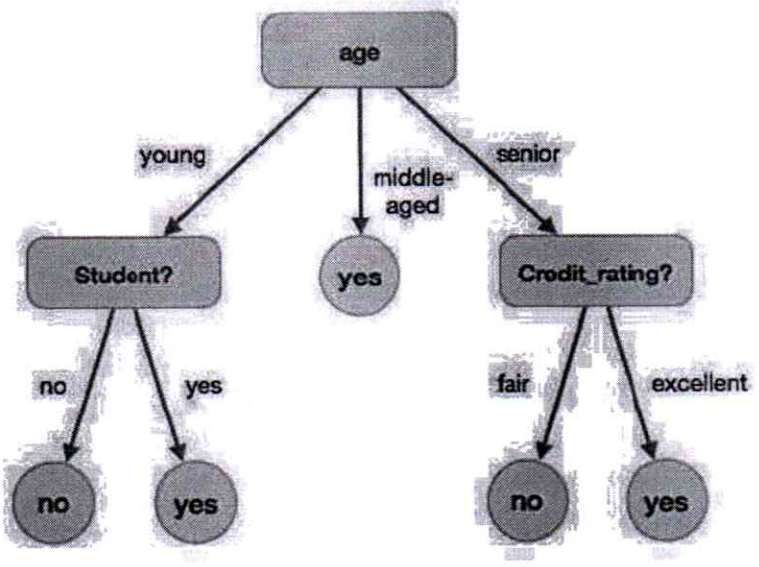
DECISION TREE

AIM

This experiment illustrates the use of Decision Tree classifier in weka. The sample data set used in this experiment is weather dataset available at arff format. This document assumes that appropriate data preprocessing has been performed.

DECISION TREE

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node. The following decision tree is for the concept buy computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class



The benefits of having a decision tree are as follows

- ✓ It does not require any domain knowledge.
- ✓ It is easy to comprehend.
- ✓ The learning and classification steps of a decision tree are simple and fast.

Decision Tree Induction Algorithm

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm, there is no backtracking; the trees are constructed in a top-down recursive divide-and-conquer manner

IT-T73 CRYPTOGRAPHY AND NETWORK SECURITY

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T73	Information Security	3	1	0

Course Objectives:

To learn about wired and wireless network security with various cryptographic techniques, which include private and public keys algorithms along with attacks types.

Course Outcomes:

On successful completion of this course students will be able to:

1. Use appropriate methods in security
2. Learn various methods of implementing security

UNIT – I CLASSICAL CRYPTOSYSTEM **9**

Security trends – Security Attacks and services – Classical Encryption Techniques — Symmetric cipher model– Basic Number theory – Pseudorandom Number Generation - Stream Ciphers - RC4.

UNIT – II BLOCK CIPHER **9**

Simple DES – DES – Modes of operation – Triple DES – AES – RSA – Attacks – Primality test – factoring.

UNIT – III MESSAGE AUTHENTICATION **9**

Discrete Logarithms – Computing discrete logs – Diffie-Hellman key exchange – ElGamal Public key cryptosystems – Hash functions – Secure Hash - MD5 – Digital signatures – RSA – ElGamal Digital signature scheme.

UNIT – IV NETWORK SECURITY **9**

Key Management and Distribution: X.509, PKI – Electronic Mail security – PGP – IP security – Web Security – SSL, TLS.

UNIT – V WIRELESS NETWORK SECURITY **9**

Wireless Network Security- IEEE 802.11 Wireless LANs - Protocol Overview and Security - Wireless Application Protocol (WAP) - Protocol Overview – Wireless Transport Layer Security (WTLS), WAP end-to-end Security

TOTAL: 45

Content beyond Syllabus:

1. Advanced techniques of security and their implementation
2. Implementation of the latest security for latest security threats

TEXT BOOKS:

1. William Stallings, "Crpyptography and Network security Principles and Practices", Pearson/PHI, 5th ed, 2006. [Unit I, Unit II, Unit IV, Unit V]
2. Wade Trappe, Lawrence C Washington, "Introduction to Cryptography with coding theory", 2nd ed, Pearson, 2007. [Unit III]

REFERENCES:

1. W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education, Second Edition, 2007.
2. Charles P. Pfleeger, Shari Lawrence Pfleeger, "Security in computing", Third Edition – Prentice Hall of India, 2006.
3. Douglas R. Stinson. "Cryptography, theory and practice" , Second edition, CRS Press.

Websites:

1. http://thor.info.uaic.ro/~fltiplea/IS/ICSCourseNotes.html
2. https://www.securityforum.org/
3. eeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=4149673

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,

IT-E66 OBJECT ORIENTED ANALYSIS AND DESIGN

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-E66	Object Oriented Analysis and Design	3	1	-

Pre-requisite:
Knowledge in the features of Object Oriented Programming Languages

Course Objectives:
To familiarize the students to carry out object oriented analysis and design for developing object oriented software projects

Course Outcomes:
Students acquire the skills to apply Industry recommended Unified Modeling Language Practices for OOAD and document them effectively

Syllabus:

Unit I

Object Oriented Methodologies: Software System Life Cycle – Traditional cycle models – Object Oriented approach – Rambaugh et al Object Modeling Technique – Booch Methodology – Jacobsonet al methodology –Rational Unified Process (RUP) – Unified Modeling Language (UML) – UML Models.

Unit II

UML Diagrams: Use case diagram – UML class diagram – interaction diagram – state diagram – activity diagram – Requirements for ATM banking system – case study.

Unit III

Object Oriented Analysis: Use case driven Object analysis – approaches for identifying classes – identifying objects, relationships attributes, methods for ATM banking system –Object oriented design process – design axioms.

Unit IV

Object Oriented Design: Designing Classes, methods – access layer object storage and object interoperability –access layer for the ATM banking system View layer – designing interface objects – prototyping User interface – view layer for the ATM banking system.

Unit V

Design Patterns: Design Patterns – Describing design patterns - catalog of design patterns – organizing the catalog – How design patterns solve design problems – How to select a design pattern – How to use a design pattern – creational pattern : Abstract factory – structural pattern : Adapter – behavioral pattern : chain of responsibility.

(Total : 45 Periods)

Content beyond Syllabus:

1. Students are encouraged to prepare the document for Mini project and Final year project applying OOAD for the system they implement.
2. Using CASE tools for performing OOAD.

Text Books:

1. Ali Bahrami, Object Oriented systems development, Tata Mcgraw Hill Education Private Ltd, 1999.
2. Carol Britton and Jill Doake, A student Gide to Object Oriented Development, Elsevier, Butterworth – Heinemann, Eighth Edition, 2007.
3. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns – elements of reusable object oriented software, Addition Wesley, 1994.

Reference Books:

1. Craig Larman,"Applying UML and Patterns: An Introduction to object-oriented Analysis and Design and iterative development", Third Edition, Pearson Education, 2005
2. Mike O'Docherty "Object-Oriented Analysis & design – understanding system development with UML 2.0", John Wiley, 2005.
3. Grady Booch, James Rumbagh, IvarJacobson, "The UML user Guide", Pearson Education, 2005
4. Timothy C. Lethbridge, Robert Laganiere" Object-Oriented Software Engineering – A practical software development using UML and Java", Tata McGraw-Hill, New Delhi, March 2003.

DECENTRALIZED BLOCKCHAIN BASED SYSTEM FOR SECURE DATA STORAGE IN CLOUD

PROJECT REPORT

Submitted by

S. DHARSHINI	**REG. NO: 17TE1205**
T. DHANYA	**REG. NO: 17TH1207**
R. GAYATHRI	**REG. NO: 17TH1212**

Under the Guidance of

Mr. S. UTHAYASHANGAR M.Tech.,

Assistant Professor

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

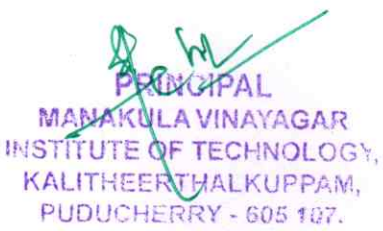
KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107

PONDICHERRY UNIVERSITY

INDIA

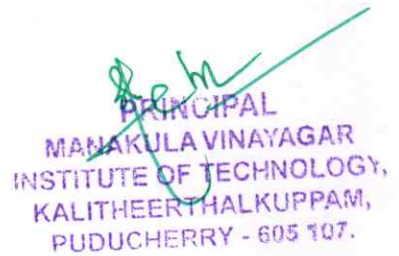
ABSTRACT

Different from the current cloud storage solutions, which are mostly centralized storage providers, this paper proposes a decentralized storage system based on blockchain technology, which can make full use of the remaining space of personal hard disks of users around the world. The blockchain is an innovative technology which opened doors to new applications for solving numerous problems in distributed environments. Cloud computing is at the top of the emerging risk list, as reported in the last survey in 2018, where 110 global executives participated. Specifically, issues predominantly related to security were addressed, such as the unauthorized access to sensitive or restricted information. Indeed, the rising level of complexity of cloud computing has made it an attractive target for attackers worldwide, who constantly monitor the security of data storage providers. In addition, senior executives are expressing mounting concerns about the possibility that cloud providers won't be able to provide access to information as a result of disruption in their own operations, causing massive downtime in their network. In the proposed system, the user's file is encrypted after split the file into multiple part and stored across multiple peers in network. This paper highly focused on decentralized secure data storage, high availability, efficient utilization of storage resources.



LIST OF ABBREVIATION

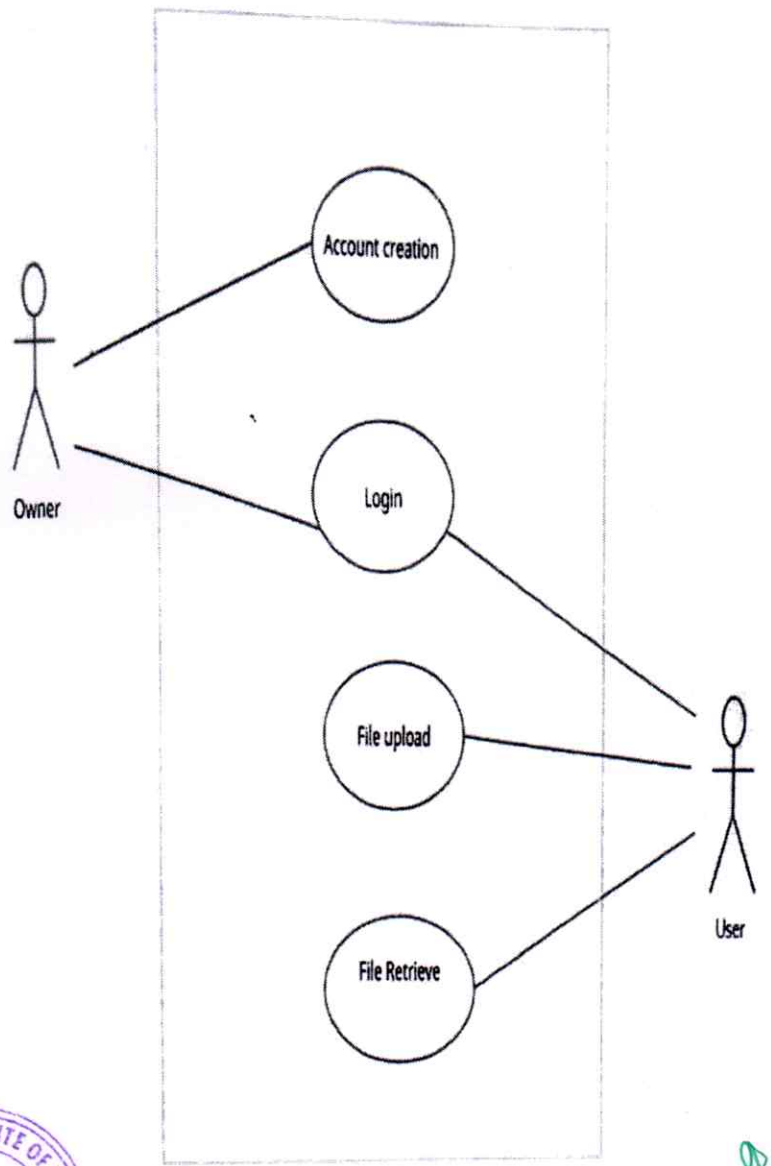
AES	Advanced Encryption Standard
SHA	Secure Hash Function
DES	Data Encryption Standard
RSA	Rivest Shamir Adleman
PAAS	Platform As A Service
SDO	Standard Development Organization
PBFT	Practical Byzantine Fault Tolerance
CP-ABE	Cipher Text Policy Attribute Based Encryption
SVGA	Super Video Graphic Adaptor
DOM	Document Object Module
IPFS	Inter Planetary File System



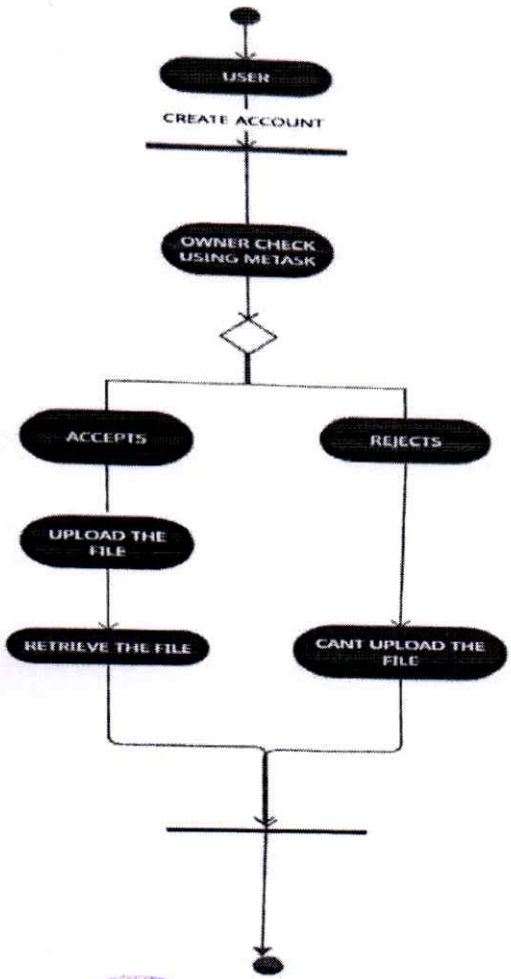
CHAPTER 6

SYSTEM DESIGN

6.1 USECASE



6.3 ACTIVITY DIAGRAM



IT-E68 USER INTERFACE DESIGN

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-E68	User Interface Design	3	1	0

Course Objectives:

1) To study the basic characteristics of graphics and web interfaces, Human Computer Interaction, multimedia interfaces for the web and the principles of evaluating interfaces.

Course Outcomes:

On successful completion of this course students will be able to:

1) The students learn concepts of user interface and used for web applications, human interfaces and for multimedia interfaces.

UNIT I

Introduction: A Taxonomy Of Software Design–Goal-Directed Design– TheThreeModels–VisualInterfaceDesign–Forms–IdiomsAndAffordances–HistoryofRectanglesontheScreen –Windows- Files– Storage and Retrieval Systems – Choosing Platforms.

UNIT II

Interface Design: BehaviorofPresentation–OrchestrationandFlow–TechniquesforInducingandMaintaining Flow–CharacteristicsofGoodUserInterface–PosturesandState–Idiocy–TheSecretWeaponofInterfaceDesign.

UNIT III

Mouse Operations: TheInteraction–MouseOperations–Selection–DirectManipulation–Manipulatinggizmos– Repositioning–Resizing and reshaping– Arrowing– Direct-Manipulation visual feedback – Drag-and-Drop.

UNIT IV

Menu Selection: The Cast– The Meaning of Menus–Menu–Dialog Boxes–DialogBox Etiquette–Toolbars–The Gizmos–Imperative and Selection Gizmos–Entry and Display Gizmos–New Gizmos.

UNIT V

Managing Exceptions & Personalization: EliminatingtheErrorMessages–ManagingExceptions–Undo–Troubles–Redo–SpecialUndo Functions– Installation–Configuration–Personalization.

(Total: 60 Periods)

Text Books:

1. Alan Cooper, The Essentials of User Interface Design,Wiley Dream techIndia (P) Ltd., 2002.
2. Ben Schneiderman, Designing theUserInterface,AddisonWesley,2000.

Reference Books:

1. AlanDix, JanetEFinlay,GregoryD.AbowdandRussellBeale, Human-ComputerInteraction,PrenticeHall,3rd Edition,2003.
2. JacobNielson, Usability Engineering, AcademicPress,1993.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

IT-T53OPERATINGSYSTEMS

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T53	Operating Systems	3	1	0

Course Objectives:

1. To grasp a fundamental understanding of operating systems
2. To learn the concepts and creation computer processes and threads
3. To understand memory management and virtual memory concepts in modern OS
4. To understand process concurrency and synchronization
5. Understand the concepts of data input/output, storage and file management
6. To learn the scheduling policies, memory management and file management of some commercial operating systems

Course Outcomes:

On successful completion of this course students will be able to:

1. Understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.,
2. Understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions,
3. Understand how the operating system abstractions can be implemented,
4. Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software,
5. Understand basic resource management techniques (scheduling or time management, space management) and principles and how they can be implemented. These also include issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection.

UnitI **(12 Periods)**

Introduction:MainframeSystems—DesktopSystems–MultiprocessorSystems–Distributed Systems–ClusteredSystems-RealTimeSystems—HardwareProtection–SystemComponents –HandheldSystems-OperatingSystemServices–SystemCalls-SystemPrograms-Process Concept–ProcessScheduling–OperationsonProcesses–CooperatingProcesses–Inter-processCommunication.

UnitII **(12 Periods)**

Threads–Overview–Threadingissues-CPUScheduling–BasicConcepts–SchedulingCriteria –SchedulingAlgorithms–Multiple-ProcessorScheduling–RealTimeScheduling-TheCritical- Section Problem –Synchronization Hardware - Semaphores– Classic problems of Synchronization –Critical regions–Monitors.

UnitIII **(12 Periods)**

System Model – Deadlock Characterization – Methods for handling Deadlocks –Deadlock Prevention – Deadlockavoidance–Deadlockdetection–RecoveryfromDeadlocks-StorageManagement –Swapping – Contiguous Memory allocation – Paging – Segmentation – Segmentation with Paging -Virtual Memory – Demand Paging – Process creation – Page Replacement–Allocation of frames–Thrashing.

UnitIV **(12 Periods)**

FileConcept–AccessMethods–DirectoryStructure–FileSystemMounting–FileSharing–Protection -FileSystemStructure–FileSystemImplementation–DirectoryImplementation–AllocationMethods-Free-spaceManagement-KernelI/OSubsystems-DiskStructure–Disk Scheduling–Disk Management–Swap-Space Management.

UnitV **(12 Periods)**

Linux overview– Kernel Architecture–Process, memory, file and I/O management –Inter- process communication and synchronization–Security - Windows XP - System Architecture – System management mechanisms – Process, thread ,memory and file management– I/O subsystem–Interprocess communication–Security

(Total : 60 Periods)

Content beyond Syllabus:

1. Introduction to Multiprocessor, Network and Distributed Operating Systems.

Text Books:

1. AbrahamSilberschatz,PeterBaerGalvinandGregGagne,OperatingSystemConcepts,Seventh

IT-E51 COMPUTER HARDWARE AND TROUBLESHOOTING

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-E51	Computer Hardware and Troubleshooting	3	1	0

Course Objectives:

1. It provides insight to the various parts and types of computer.
2. It familiarizes the hardware types and the evolution in each of them.
3. It also gives the basics of troubleshooting.

Course Outcomes:

On successful completion of this course students will be able to:

- The students will have theoretical exposure as well as hands on exposure to know about the hardware aspects of computer.

Unit I

PC Hardware Overview: Introduction–BasicPartsofPC–Functionalblockdiagram–systemboard–Microprocessor– Interrupts–DMA–SMPS–BIOS–POSTsequence-Systemconfigurationion–Memory–Massstorage–I/Ointerfacestandards.

Unit II

Bus Standards and Networking: ISA–PCI–SCSI–IDE–USB–comparativestudyandcharacteristics–NetworkInterface Cards–Cables and connectors-MODEM–AT command set.

Unit III

Peripheral Devices & Display Adapters: Functional descriptions of keyboard – mouse – printers – joystick – scanners – CGA – SVGA.

Unit IV

Mass Storage Devices: Floppy disk and drive – Hard disk and drive – MFM and RLL recording standards – CD technology – DVD technology – pen drives – tape drives.

Unit V

Troubleshooting tools: In-CircuitEmulators–LogicState/TimingAnalyzers–DigitalMultimeters–CROs–Signature Analyzers–Troubleshooting problems of system boards ,add oncards and peripherals.

(Total: 60 Periods)

Content beyond Syllabus:

1. Advanced data structures and their implementation
2. Implementation of the data structures in different language platforms

Text Books:

1. Hans Peter Messmer, Indispensable PC Hardware Book , Pearson Education, 4th edition,2003.
2. Govindarajulu, IBM PC and Clones , Tata McGraw Hill, 4th edition, 2002.

Reference Books:

1. Barry Brey, The Intel Microprocessors 8086/88, 80186/188, 80286, 80386,80486, PENTIUM and PENTIUM PRO architecture, Programming and Interfacing, 6th edition, PHI, 2002.
2. Ed Tittel, David Johnson, Networking Essentials: Study Guide, Comdex Computer Publishing,1998.
3. Scott Muller, Upgrading and Repairing PCs, 15th edition, 2002.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

HOUSEHOLD ELECTRICITY POWER CONSUMPTION PREDICTION USING CNN-GRU TECHNIQUES

PROJECT REPORT

PHASE- II

Submitted by

DIVYABHARATHI.J	**REG. NO: 17TH1209**
HARSHAVARDENI.D	**REG. NO: 17TH1215**
MEENA.S	**REG. NO: 17TH1234**

Under the Guidance of

Mr. M. SURESH, M. Tech., (Ph.D)

Assistant Professor

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY-605 107

PONDICHERRY UNIVERSITY

INDIA

MAY - 2021

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY * PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

- **Hard Disk** : 500GB
- **Monitor** : 15 VGA color
- **Mouse** : Any Optical Mouse
- **Keyboard** : 110 keys enhanced
- **RAM** : 4GB(min)

3.2 SOFTWARE REQUIREMENTS

- **Operating System** : Linux/Windows
- **Language** : Python
- **Tool** : PyCharm

3.3 ABOUT THE SOFTWARE

For this purpose, Linux/Windows operating system will be needed and python programming language is used and the deep learning technique will be integrated with the python program and it will be implemented in the PyCharm software.

- **Linux**

Linux is a Unix-like, open source and community-developed operating system for computers, servers, mainframes, mobile devices and embedded devices. It is supported on almost every major computer platform including x86, ARM and SPARC, making it one of the most widely supported operating systems.

How is Linux operating system used?

Every version of the Linux operating system manages hardware resources, launches and handles applications, and provides some form of user interface. The enormous development community and wide range of distributions means that a Linux version is available for almost any task, and Linux has penetrated many areas of computing.

For example, Linux has emerged as a popular operating system for web servers such as Apache, as well as for network operations, scientific computing tasks that require huge compute clusters, running databases, desktop/endpoint computing and running mobile devices with OS versions like Android.

Linux distributions

Since its initial development, Linux has adopted the copyleft stipulations of the Free Software Foundation which originated the GNU GPL General Public License (GPL). Copyleft says that anything taken for free and modified must in turn be distributed for free. In practice, if Linux or other GNU components are developed or modified to create a new version of Linux, that new version must be distributed for free. This is the foundation of open source development which prevents a developer or other groups from profiting from the freely available work of others.

The GNU GPL does not prohibit intellectual ownership, and it is commonplace for creators of Linux components to hold copyrights on the various components. The GNU GPL ensures that those components remain free and freely distributed. While the software remains free, however, it is common for some commercial distributions to charge for value -added services, such as support or custom development services.

Linux components

The Linux operating system follows a modular design that is the key to its many variations and distributions. A bootloader is responsible for starting the Linux kernel. The kernel is at the core of the Linux system, handling network access, scheduling processes or applications, managing basic peripheral devices, and overseeing file system services.

But it is really the many outside developers and GNU projects that offer high-level functions to the Linux kernel to provide a fully realized operating system. For example, there are modules

to provide a command line interface, implement a graphical user interface, manage security, offer video input or audio services and many others -- each of which can be modified and optimized to form unique distributions for specific tasks.

Package manager software typically adds, updates or removes software components under the Linux operating system. Examples of package managers include dpkg, OpenPKG, RPM Package Manager and Zero Install.

History of Linux

Linus Torvalds started working on Linux as a replacement to the MINIX operating system while at the University of Helsinki in Finland. Torvalds recognized the work done on the GNU Project in 1983, which intended to create a complete, Unix-compatible operating system comprised entirely of free software, and noted the GNU as a model for distribution. However, the work on GNU had not been finished by the time Torvalds sought a MINIX replacement, prompting him to develop an alternate operating system kernel dubbed Linux -- a contraction of "Linus' Unix" -- and adopt the GNU GPL.

- **Windows**

Microsoft Windows, commonly referred to as **Windows**, is a group of several proprietary graphical operating system families, all of which are developed and marketed by Microsoft. Each family caters to a certain sector of the computing industry. Active Microsoft Windows families include Windows NT and Windows IoT; these may encompass subfamilies, e.g. Windows Server or Windows Embedded Compact (Windows CE). Defunct Microsoft Windows families include Windows 9x, Windows Mobile and Windows Phone.

Microsoft introduced an operating environment named *Windows* on November 20, 1985, as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs).[4] Microsoft Windows came to dominate the world's personal computer (PC) market with over 90% market share, overtaking Mac OS, which had been introduced in 1984. Apple came to see Windows as an unfair encroachment on their innovation in GUI development as implemented on products such as the Lisa and Macintosh (eventually settled in court in Microsoft's favor in 1993). On PCs, Windows is still the most popular operating system. However, in 2014, Microsoft admitted losing the majority of the overall operating system market

IT-T73 CRYPTOGRAPHY AND NETWORK SECURITY

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T73	Information Security	3	1	0

Course Objectives:

To learn about wired and wireless network security with various cryptographic techniques, which include private and public keys algorithms along with attacks types.

Course Outcomes:

On successful completion of this course students will be able to:

1. Use appropriate methods in security
2. Learn various methods of implementing security

UNIT – I CLASSICAL CRYPTOSYSTEM **9**

Security trends – Security Attacks and services – Classical Encryption Techniques — Symmetric cipher model– Basic Number theory – Pseudorandom Number Generation - Stream Ciphers - RC4.

UNIT – II BLOCK CIPHER **9**

Simple DES – DES – Modes of operation – Triple DES – AES – RSA – Attacks – Primality test – factoring.

UNIT – III MESSAGE AUTHENTICATION **9**

Discrete Logarithms – Computing discrete logs – Diffie-Hellman key exchange – ElGamal Public key cryptosystems – Hash functions – Secure Hash - MD5 – Digital signatures – RSA – ElGamal Digital signature scheme.

UNIT – IV NETWORK SECURITY **9**

Key Management and Distribution: X.509, PKI – Electronic Mail security – PGP – IP security – Web Security – SSL, TLS.

UNIT – V WIRELESS NETWORK SECURITY **9**

Wireless Network Security- IEEE 802.11 Wireless LANs - Protocol Overview and Security - Wireless Application Protocol (WAP) - Protocol Overview – Wireless Transport Layer Security (WTLS), WAP end-to-end Security

TOTAL: 45

Content beyond Syllabus:

1. Advanced techniques of security and their implementation
2. Implementation of the latest security for latest security threats

TEXT BOOKS:

1. William Stallings, "Crpyptography and Network security Principles and Practices", Pearson/PHI, 5th ed, 2006. [Unit I, Unit II, Unit IV, Unit V]
2. Wade Trappe, Lawrence C Washington, "Introduction to Cryptography with coding theory", 2nd ed, Pearson, 2007. [Unit III]

REFERENCES:

1. W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education, Second Edition, 2007.
2. Charles P. Pfleeger, Shari Lawrence Pfleeger, "Security in computing", Third Edition – Prentice Hall of India, 2006.
3. Douglas R. Stinson. "Cryptography, theory and practice" , Second edition, CRS Press.

Websites:

1. http://thor.info.uaic.ro/~fltiplea/IS/ICSCourseNotes.html
2. https://www.securityforum.org/
3. eeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=4149673

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,

IT-E66 OBJECT ORIENTED ANALYSIS AND DESIGN

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-E66	Object Oriented Analysis and Design	3	1	-

Pre-requisite:
Knowledge in the features of Object Oriented Programming Languages

Course Objectives:
To familiarize the students to carry out object oriented analysis and design for developing object oriented software projects

Course Outcomes:
Students acquire the skills to apply Industry recommended Unified Modeling Language Practices for OOAD and document them effectively

Syllabus:

Unit I

Object Oriented Methodologies: Software System Life Cycle – Traditional cycle models – Object Oriented approach – Rambaugh et al Object Modeling Technique – Booch Methodology – Jacobsonet al methodology –Rational Unified Process (RUP) – Unified Modeling Language (UML) – UML Models.

Unit II

UML Diagrams: Use case diagram – UML class diagram – interaction diagram – state diagram – activity diagram – Requirements for ATM banking system – case study.

Unit III

Object Oriented Analysis: Use case driven Object analysis – approaches for identifying classes – identifying objects, relationships attributes, methods for ATM banking system –Object oriented design process – design axioms.

Unit IV

Object Oriented Design: Designing Classes, methods – access layer object storage and object interoperability –access layer for the ATM banking system View layer – designing interface objects – prototyping User interface – view layer for the ATM banking system.

Unit V

Design Patterns: Design Patterns – Describing design patterns - catalog of design patterns – organizing the catalog – How design patterns solve design problems – How to select a design pattern – How to use a design pattern – creational pattern : Abstract factory – structural pattern : Adapter – behavioral pattern : chain of responsibility.

(Total : 45 Periods)

Content beyond Syllabus:

1. Students are encouraged to prepare the document for Mini project and Final year project applying OOAD for the system they implement.
2. Using CASE tools for performing OOAD.

Text Books:

1. Ali Bahrami, Object Oriented systems development, Tata Mcgraw Hill Education Private Ltd, 1999.
2. Carol Britton and Jill Doake, A student Gide to Object Oriented Development, Elsevier, Butterworth – Heinemann, Eighth Edition, 2007.
3. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns – elements of reusable object oriented software, Addition Wesley, 1994.

Reference Books:

1. Craig Larman,"Applying UML and Patterns: An Introduction to object-oriented Analysis and Design and iterative development", Third Edition, Pearson Education, 2005
2. Mike O'Docherty "Object-Oriented Analysis & design – understanding system development with UML 2.0", John Wiley, 2005.
3. Grady Booch, James Rumbagh, IvarJacobson, "The UML user Guide", Pearson Education, 2005
4. Timothy C. Lethbridge, Robert Laganiere" Object-Oriented Software Engineering – A practical software development using UML and Java", Tata McGraw-Hill, New Delhi, March 2003.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY

DECENTRALIZED BLOCKCHAIN BASED SYSTEM FOR SECURE DATA STORAGE IN CLOUD

PROJECT REPORT

Submitted by

S. DHARSHINI	**REG. NO: 17TE1205**
T. DHANYA	**REG. NO: 17TH1207**
R. GAYATHRI	**REG. NO: 17TH1212**

Under the Guidance of

Mr. S. UTHAYASHANGAR M.Tech.,

Assistant Professor

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107

PONDICHERRY UNIVERSITY

INDIA

MAY – 2021

ABSTRACT

Different from the current cloud storage solutions, which are mostly centralized storage providers, this paper proposes a decentralized storage system based on blockchain technology, which can make full use of the remaining space of personal hard disks of users around the world. The blockchain is an innovative technology which opened doors to new applications for solving numerous problems in distributed environments. Cloud computing is at the top of the emerging risk list, as reported in the last survey in 2018, where 110 global executives participated. Specifically, issues predominantly related to security were addressed, such as the unauthorized access to sensitive or restricted information. Indeed, the rising level of complexity of cloud computing has made it an attractive target for attackers worldwide, who constantly monitor the security of data storage providers. In addition, senior executives are expressing mounting concerns about the possibility that cloud providers won't be able to provide access to information as a result of disruption in their own operations, causing massive downtime in their network. In the proposed system, the user's file is encrypted after split the file into multiple part and stored across multiple peers in network. This paper highly focused on decentralized secure data storage, high availability, efficient utilization of storage resources.

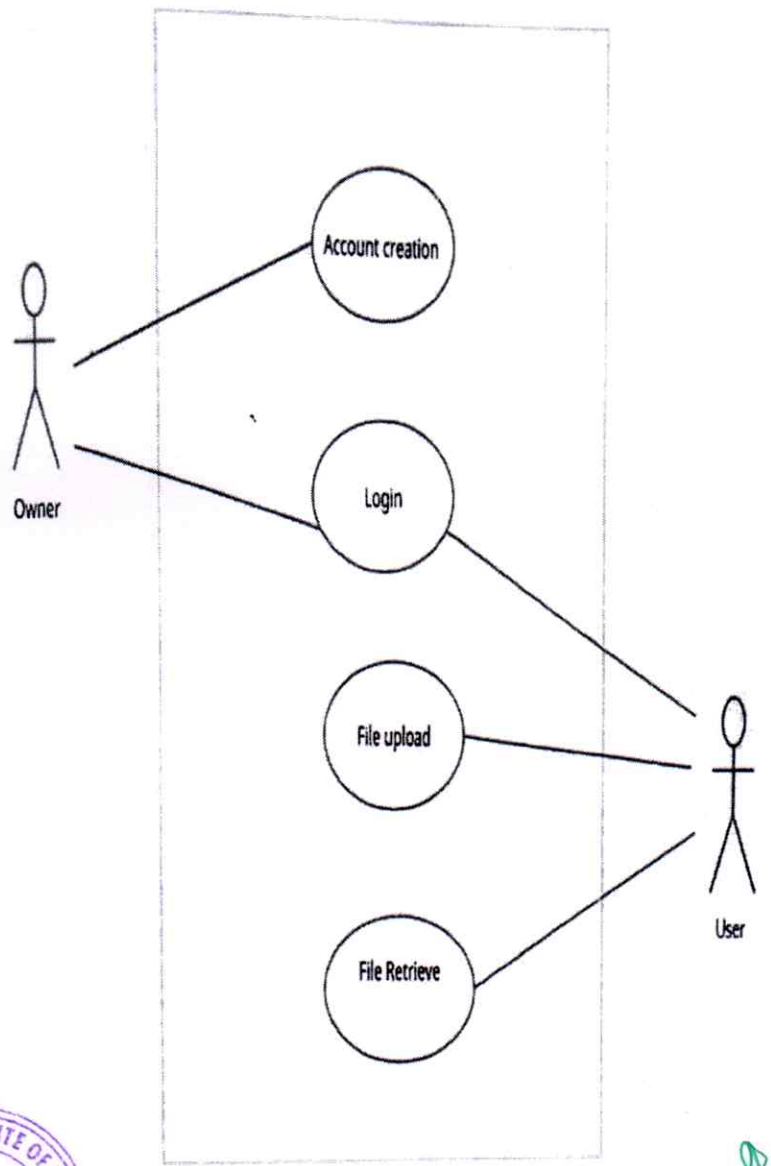
LIST OF ABBREVIATION

AES	Advanced Encryption Standard
SHA	Secure Hash Function
DES	Data Encryption Standard
RSA	Rivest Shamir Adleman
PAAS	Platform As A Service
SDO	Standard Development Organization
PBFT	Practical Byzantine Fault Tolerance
CP-ABE	Cipher Text Policy Attribute Based Encryption
SVGA	Super Video Graphic Adaptor
DOM	Document Object Module
IPFS	Inter Planetary File System

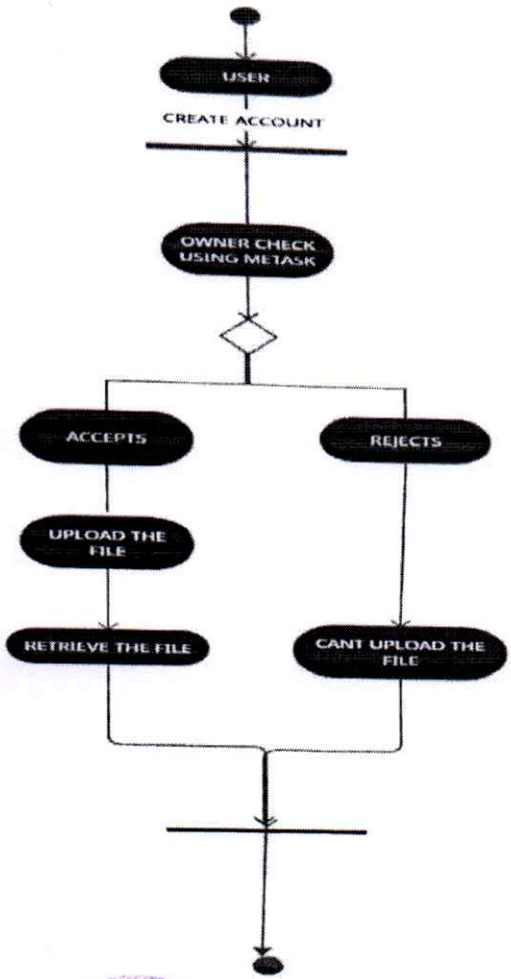
CHAPTER 6

SYSTEM DESIGN

6.1 USECASE



6.3 ACTIVITY DIAGRAM



IT-T63 ARTIFICIAL INTELLIGENCE

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T63	Artificial Intelligence	3	1	0

Pre-requisite:
Knowledge in Programming, Discrete mathematics and in probability.

Course Objectives:

1. To search and discover intelligent characteristics of existing AI projects, Intelligent agents map a new problem – as search.
2. To understand different search strategies for a problem.
3. To understand different Knowledge Representation schemes for typical AI problems.
4. To design and implement a typical AI problem to be solved Using Machine Learning Techniques.
5. Implement a futuristic AI application.

Course Outcomes:

On successful completion of this course students will be able to:

1. Capability to develop intelligent systems
2. Apply heuristic concepts to design efficient algorithms that help to attain the goals in satisfactory manner
3. Design applications related to Natural Language Processing and Web applications.

Unit I

Introduction: History of AI - - problem spaces and search- Heuristic Search techniques –Best-first search- Problem reduction-Constraint satisfaction-Means Ends Analysis.
Intelligent agents: Agents and environment – structure of agents and its functions

Unit II

Knowledge Representation: Approaches and issues in knowledge representation- Propositional Logic –Predicate logic-Forward and backward reasoning - Unification- Resolution- Weak slot-filler structure – Strong slot-filler structure- Knowledge- Based Agent

Unit III

Reasoning under uncertainty: Logics of non-monotonic reasoning-Implementation- Basic probability notation - Bayes rule – Certainty factors and rule based systems-Bayesian networks – Dempster - Shafer Theory - Fuzzy Logic.

Unit IV

Planning and Learning: Planning with state space search-partial order planning-planning graphs-conditional planning-continuous planning-Multi-Agent planning. Forms of learning-inductive learning-learning decision trees-ensemble learning-Neural Net learning and Genetic learning

Unit V

Advanced Topics: Game Playing: Minimax search procedure-Adding alpha-beta cutoffs Expert System: Representation-Expert System shells-Knowledge Acquisition. Robotics: Hardware-Robotic Perception-Planning-Application domains

(Total : 60 Periods)

Content beyond Syllabus:

1. Natural language understanding and generation.
2. Speech processing system.

Text Books:

1. Elaine Rich and Kevin Knight and ShivashankarB.Nair, Artificial Intelligence, 3rd edition, Tata Mc Graw Hill, 2009.
2. Ben Coppin, "Artificial Intelligence Illuminated", Jones and Bartlett Publishers, 1st edition, 2004.
3. Stuart J.Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson Education Asia, II edition, 2003.
4. N.P.Padhy, Artificial Intelligence and Intelligent Systems, Oxford University Press, 2nd edition, 2005.

Reference Books:

1. RajendraAkerkar ,Introduction to Artificial Intelligence, Prentice hall of India, 2005.
2. Patrick Henry Winston, Artificial Intelligence, 3rd edition Pearson Education, Inc., 2001.

Websites:

1. http://aima.cs.berkeley.edu/ai.html
2. www.stanford.edu/**class**/cs221/

IMAGE TEXT TO SPEECH CONVERSION USING DEEP LEARNING TECHNIQUES

PROJECT REPORT

Submitted by

R. AMUDHA — **REGISTER NO: 17TH1203**

J. ANUSIYA — **REGISTER NO: 17TH1204**

K. HASHVANDHINI — **REGISTER NO: 17TH1216**

Under the Guidance of

Mr. K.VIJIYAKUMAR M.Tech., [Ph. D]

Assistant Professor

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107

PONDICHERRY UNIVERSITY

INDIA

MAY – 2021

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY PUDUCHERRY 605 107.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability allowing it to learn from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy. Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services such as digital assistants, voice-enabled TV remotes, and credit card fraud detection as well as emerging technologies such as self-driving cars. Deep learning is a new emerging area which exploits artificial intelligence and machine learning to learn features directly from the data, using multiple nonlinear processing layers. Deep learning models can achieve very high accuracy in email spam classification. The fundamental constituent of deep learning is the multilayered hierarchical data representation typically in the form of a neural network with more than two layers.

1.2 WHAT IS DEEP LEARNING?

Deep learning, also known as deep neural networks or neural learning, is a form of artificial intelligence (AI) that seeks to replicate the workings of a human brain. It is a form of machine learning, with functions that operate in a nonlinear decision-making process. Deep learning occurs when decisions are made on unstructured data without supervision. Object recognition, speech recognition, and language translation are some of the tasks performed through deep learning.

1.3 DEEP LEARNING VS MACHINE LEARNING

One of the most common AI techniques used for processing big data is machine learning, a self-adaptive algorithm that gets increasingly better analysis and patterns with experience or with newly added data.

process in which a model learns to become more accurate for performing an action in an environment based on feedback in order to maximize the reward.

1.4 CHARACTERISTICS OF DEEP LEARNING

➢ **Supervised, Semi-Supervised or Unsupervised**

When the category labels are present while you train the data then it is Supervised learning. Algorithms like Linear regression. Logistic regression, decision trees use Supervised Learning. When category labels are not known while you train data then it is unsupervised learning. Algorithms like Cluster Analysis, K means clustering, Anomaly detection uses Unsupervised Learning in such Figure 1.2

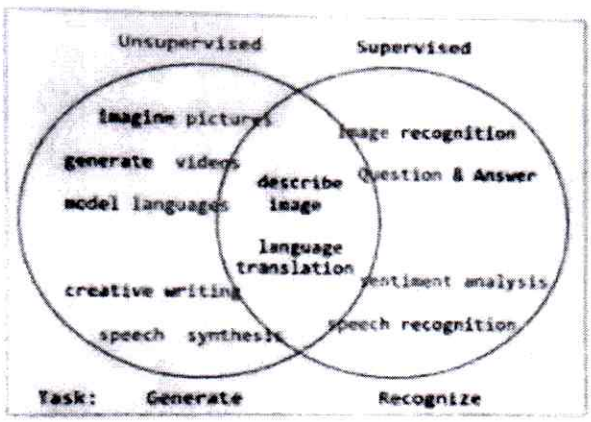


Figure 1.2 Supervised and Unsupervised

The data set consists of both labeled and unlabeled data then we call it is Semi-Supervised learning. Graph-based models, Generative models, cluster assumption, continuity assumption use Semi-Supervised learning.

➢ **Huge Amount of Resources**

It needs advanced Graphical Processing Units for processing heavy workloads. A huge amount of data needs to be processed like Big data in the form of structured or unstructured data. Sometimes more time also required to process the data, it depends on the amount of data fed in.

2.2.3 OCR BASED IMAGE TEXT TO SPEECH CONVERSION USING MATLAB

Author: Sneha.C.Madre.

Description: There are millions of blind people in the world who are visually impaired. Disability to read has a large impact on the life of visually impaired people. The Proposed system is cost-efficient and helps the visually impaired person to hear the text. The main idea of this project is optical Character recognition which is used to convert text character into the audio signal. The text is preprocessed and then used for recognition by segmenting each character. Segmentation is followed by extraction of the letter and resizing of the file containing the text. This Text file is then converted into the audio signal.MATLAB16 will be used for all these processes mentioned above.

2.2.4 IMAGE TEXT TO SPEECH CONVERSION USING RASPBERRY PI

Author: K.NimalaKumari, Meghna Reddy.

Description: In this paper an innovative, efficient and real-time cost beneficial technique that enables user to hear the contents of text images instead of reading through them as been introduced. It combines the concept of Optical Character Recognition (OCR) and Text to Speech Synthesizer (TTS) in Raspberry pi. This paper describes the design, implementation and experimental results of the device. This device consists of two modules, image processing module and voice processing module. The device was developed based on Raspberry Pi v2 with 900 MHz processor speed.

2.2.5 TEXT TO SPEECH CONVERSION

Author: S. Venkateswarlu1, K. Kamesh1,R. Sastry

Description: The present paper has introduced an innovative, efficient and real-time cost beneficial technique that enables user to hear the contents of text images instead of reading through them. This paper describes the design, implementation and experimental results of the device. This device consists of two modules, image processing module and voice processing module. The device was developed based on Raspberry Pi v2 with 900 MHz processor speed.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

- Desktop / Laptop with 4GB Ram

3.2 SOFTWARE REQUIREMENTS

- Pycharm IDE
- Python Compiler
- Google-Text-to-Speech conversion
- GoogleTrans
- Python Imaging Library
- Optical Character Recognition

3.3 ABOUT THE SOFTWARE

➢ OCR

OCR (Optical Character Recognition) has played important role in this module. OCR is a technology which is used to recognize text from printed scanned documents through the optical mechanism. Tesseract is an optical character recognition engine. To install it type "pip install pytesseract" command in the command prompt. OCR software is used to convert image into text format. It is a conversion of image of typed or handwritten or printed text into machine encoded text. It is use for blind and visually impaired people also used for automatic number plate recognition. Tesseract is a type of OCR engine with matrix matching. The selection of Tesseract engine is due to its flexibility and extensibility of machines and the fact that many communities are active researchers to develop this OCR engine and also due to this reason Tesseract OCR can support more than 100 languages. In this project we are identifying 120 to 150 languages.

IT-T71 MOBILE COMPUTING

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T71	Mobile Computing	3	1	0

Course Objectives:

1. To teach the basics of mobile computing ideas and best practices.
2. To teach the emerging wireless network standards.
3. To introduce the various models and data management concepts of mobile computing.
4. To learn the routing and secure protocols of mobile networking.

Course Outcomes:

On successful completion of the module students will be able to:

1. Gain basic knowledge in mobile computing.
2. Should have a broader knowledge on 3G.
3. Gain the knowledge on emerging wireless network standards.

Syllabus:

Unit I

INTRODUCTION: Wireless and Mobile Computing Architecture – Limitations of wireless and mobile communication – Wireless Telecommunication Networks: Digital cellular Systems, TDMA - CDMA – Wireless Networking Techniques – Mobility Bandwidth Tradeoffs – Portable Information Appliances.

Unit II

EMERGING WIRELESS NETWORK STANDARDS: 3 G Wireless Networks – State of Industry – Mobility support Software – End User Client Application – Mobility Middleware –Middleware for Application Development - Adaptation and Agents - Service Discovery Middleware – Finding Needed Services - Interoperability and Standardization.

Unit III

MOBILE NETWORKING: Virtual IP Protocols - Loose Source Routing Protocols - Mobile IP – CDPD – GPRS – UMTS Security and Authentication – Quality of Service – Mobile Access to the World Wide Web.

Unit IV

MOBILE DATA MANAGEMENT: Mobile Transactions - Reporting and Co Transactions –Kangaroo Transaction Model – Clustering Model –Isolation only transaction – 2 Tier Transaction Model – Semantic based nomadic transaction processing.

Unit V

MOBILE COMPUTING MODELS: Client Server model – Client/Proxy/Server Model – Disconnected Operation Model – Mobile Agent Model – Thin Client Model – Tools: Java, Brew, Windows CE, WAP, Sybian, and EPOC.

(Total: 60 Periods)

Content beyond the Syllabus:

1. Data dissemination and broadcasting systems.
2. Data synchronization in mobile computing systems.

Text Books:

1. Reza B Fat and Roy.T. Fielding, "Mobile Computing Principles", Cambridge University Press, 2005.
2. Abdelsalam A Helal, Richard Brice, Bert Haskel, MarekRusinkiewicz, Jeffery L Caster and DarellWoelk, "Anytime, Anywhere Computing, Mobile Computing Concepts and Technology", Springer International Series in Engineering and Computer Science, 2000.

Reference Books:

1. Golden Richard, Frank Adelstein, Sandeep KS Gupta, Golden Richard and Loren Schwiebert, "Fundamentals of Mobile and Pervasive Computing", McGraw-Hill Professional Publishing", 2005.
2. UweHansmann, LotharMerk, Martin S. Nicklons and Thomas Stober, "Principles of Mobile Computing", Springer, 2003.

Websites:

1. http://www.faadooengineers.com/threads/394-MOBILE-COMPUTING-E-book presentation-and-lecture-notes-covering-full-semester-syllabus
   http://www.dsc.ufcg.edu.br/~sampaio/cursos/2005.1/BancoDeDados/Artigos/BDMoveis/MobileTransactions/an-overview-of-transaction.pdf

IT-E66 OBJECT ORIENTED ANALYSIS AND DESIGN

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-E66	Object Oriented Analysis and Design	3	1	-

Pre-requisite:
Knowledge in the features of Object Oriented Programming Languages

Course Objectives:
To familiarize the students to carry out object oriented analysis and design for developing object oriented software projects

Course Outcomes:
Students acquire the skills to apply Industry recommended Unified Modeling Language Practices for OOAD and document them effectively

Syllabus:

Unit I

Object Oriented Methodologies: Software System Life Cycle – Traditional cycle models – Object Oriented approach – Rambaugh et al Object Modeling Technique – Booch Methodology – Jacobsonet al methodology –Rational Unified Process (RUP) – Unified Modeling Language (UML) – UML Models.

Unit II

UML Diagrams: Use case diagram – UML class diagram – interaction diagram – state diagram – activity diagram – Requirements for ATM banking system – case study.

Unit III

Object Oriented Analysis: Use case driven Object analysis – approaches for identifying classes – identifying objects, relationships attributes, methods for ATM banking system –Object oriented design process – design axioms.

Unit IV

Object Oriented Design: Designing Classes, methods – access layer object storage and object interoperability –access layer for the ATM banking system View layer – designing interface objects – prototyping User interface – view layer for the ATM banking system.

Unit V

Design Patterns: Design Patterns – Describing design patterns - catalog of design patterns – organizing the catalog – How design patterns solve design problems – How to select a design pattern – How to use a design pattern – creational pattern : Abstract factory – structural pattern : Adapter – behavioral pattern : chain of responsibility.

(Total : 45 Periods)

Content beyond Syllabus:

1. Students are encouraged to prepare the document for Mini project and Final year project applying OOAD for the system they implement.
2. Using CASE tools for performing OOAD.

Text Books:

1. Ali Bahrami, Object Oriented systems development, Tata Mcgraw Hill Education Private Ltd, 1999.
2. Carol Britton and Jill Doake, A student Gide to Object Oriented Development, Elsevier, Butterworth – Heinemann, Eighth Edition, 2007.
3. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns – elements of reusable object oriented software, Addition Wesley, 1994.

Reference Books:

1. Craig Larman,"Applying UML and Patterns: An Introduction to object-oriented Analysis and Design and iterative development", Third Edition, Pearson Education, 2005
2. Mike O'Docherty "Object-Oriented Analysis & design – understanding system development with UML 2.0", John Wiley, 2005.
3. Grady Booch, James Rumbagh, IvarJacobson, "The UML user Guide", Pearson Education, 2005
4. Timothy C. Lethbridge, Robert Laganiere" Object-Oriented Software Engineering – A practical software development using UML and Java", Tata McGraw-Hill, New Delhi, March 2003.

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY

FARMER HELPER MOBILE APPLICATION USING ANDROID

PROJECT REPORT

Submitted by

M.ABINAYA	**REGISTER NO: 17TH1201**
K.FAZAR	**REGISTER. NO: 17TH1210**
B.SWATHI	**REGISTER. NO: 17TH1251**

Under the Guidance of

Dr. P.SIVAKUMAR M.E., Ph. D

Professor and Head

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107

PONDICHERRY UNIVERSITY

INDIA

MAY – 2021

management, and other underlying functionality. Using a Linux kernel enables Android to take advantage of Linux-based security features and allows device manufacturers to develop hardware drivers for a well-known kernel.

3.3.2 Java

Current versions of Android use the latest Java language and its libraries (but not full graphical user interface (GUI) frameworks), not the Apache Harmony Java implementation, that older versions used. Java 8 source code that works in latest version of Android, can be made to work in older versions of Android.While most Android applications are written in Java-like language, there are some differences between the Java API and the Android API, and Android does not run Java bytecode by a traditional Java virtual machine (JVM), but instead by a Dalvik virtual machine in older versions of Android, and an Android Runtime (ART) in newer versions, that compile the same code that Dalvik runs to Executable and Linkable Format (ELF) executables containing machine code.Java bytecode in Java Archive (JAR) files is not executed by Android devices. Instead, Java classes are compiled into a proprietary bytecode format and run on Dalvik (or compiled version thereof with newer ART), a specialized virtual machine (VM) designed for Android. Unlike Java VMs, which are stack machines (stack-based architecture), the Dalvik VM is a register machine (register-based architecture).

3.3.3 Android Studio

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.Android offers a unified approach to application development for mobile devices which means developers need to develop only for Android, and their applications should be able to run on different devices powered by Android.Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as Google Play or the Amazon Appstore. Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and is growing fast. Every day more than 1 million new Android devices are activated worldwide.

CHAPTER 5

PROPOSED SYSTEM

5.1 PROPOSED WORK

The proposed system is done mainly to avoid middle man between the farmers and buyers. Here the farmers can fix the prices for the crops. The main objective is to sell the crops in wholesale. The farmers will also get a suggestion about the location which has type of the crops. And the current schemes produced by the government are also provided for the farmers.

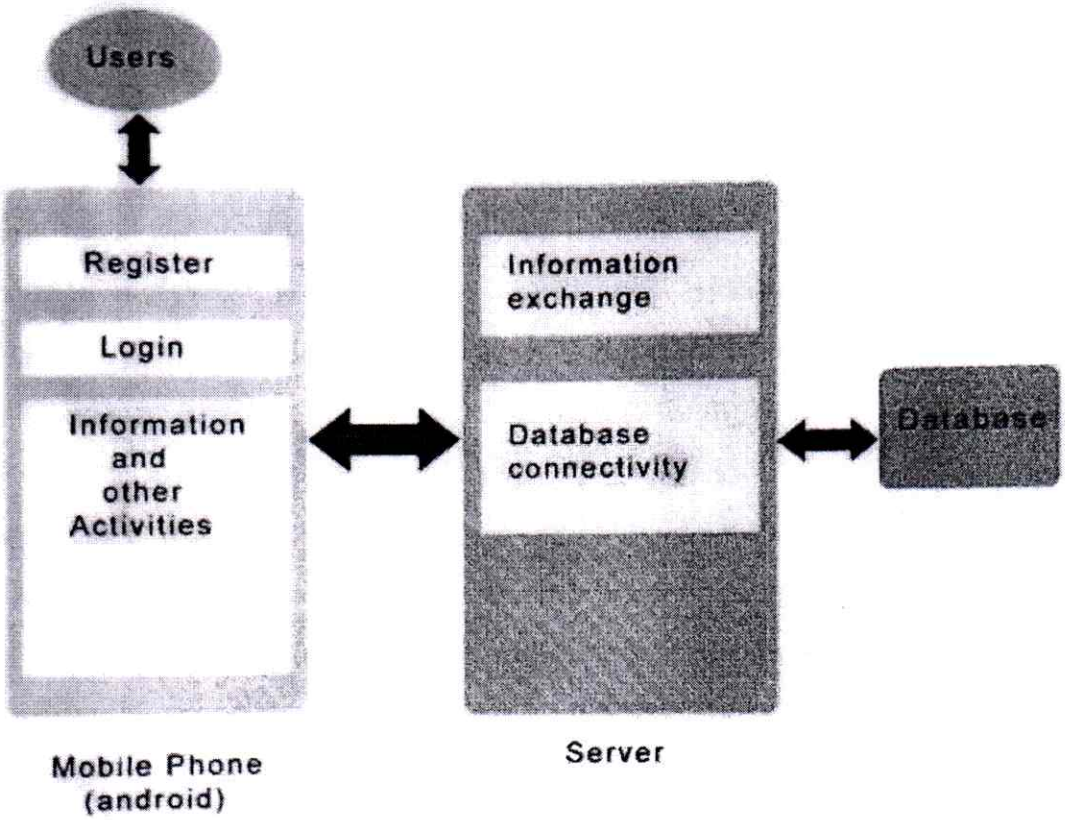


FIG 5.1 AGRICULTURE HELPER SYSTEM ARCHITECTURE

5.2. HOME SCREEN

Home Screen is the main page of this Application. This module shows some category options to the user 80 to their choice. By clicking someone option user get maisn page of that particular category which is easy to use by the user. It shows three type options,

- GPS
- Market Rates
- Government Schemes

CHAPTER 6

SYSTEM DESIGN AND ANALYSIS

6.1 USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases. A Use Case describes a sequence of actions that provided something of unmeasurable value to an actor and is drawn as a horizontal ellipse. An actor is a person, organization or external system that plays a role in one or more interaction with the system.

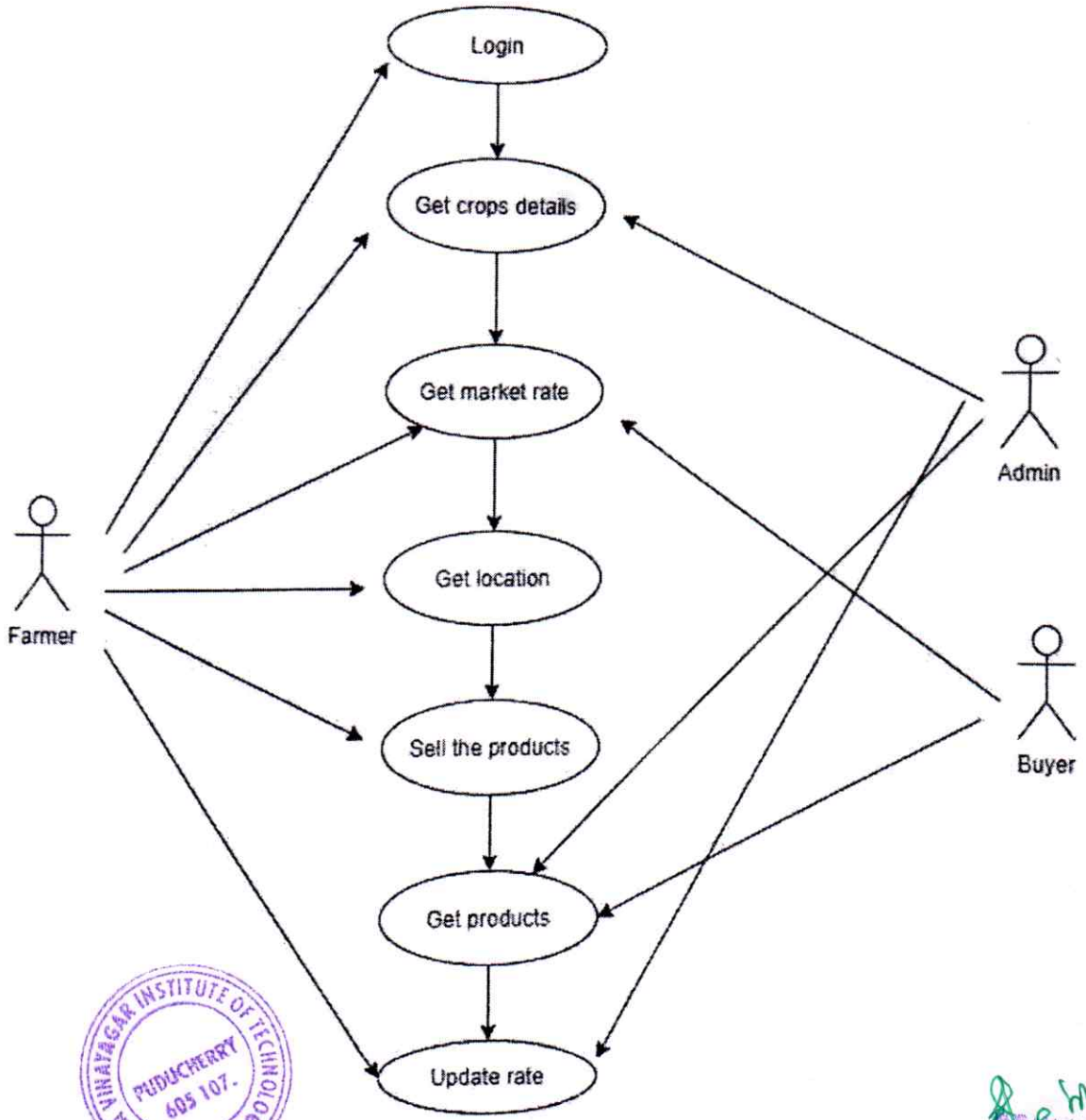


FIG 6.1 USE CASE DIAGRAM

6.2 SEQUENCE DIAGRAM

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

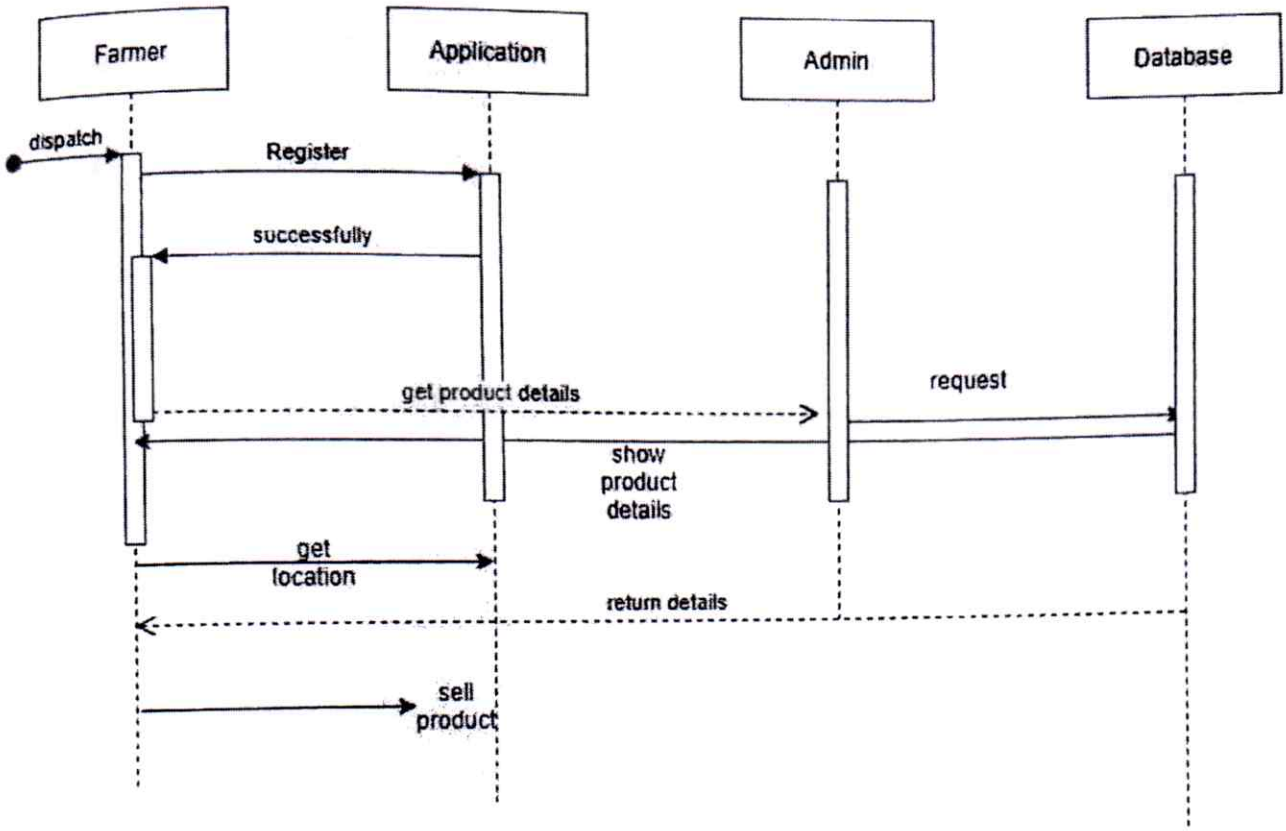


FIG 6.2 SEQUENCE DIAGRAM

6.3 ACTIVITY DIAGRAM

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

IT-T44 MICROPROCESSORS AND MICROCONTROLLERS

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T44	Microprocessors and Microcontrollers	3	1	0

Pre-requisite:
Digital Electronics and Computer Architecture

Course Objectives:

1. To understand the architectures and the instruction set of 8085 microprocessor
2. To understand the architectures and the instruction set of 8086 microprocessor
3. To understand the architectures and the instruction set of 8051 microcontroller
4. To learn the assembly language program using 8085, 8086 and 8051 instruction set
5. To learn interfacing of microprocessors and microcontrollers with various peripheral

Course Outcomes:

On successful completion of this course students will be able to:

1. Understanding the inner working components of the microprocessor and microcontrollers
2. Developing assembly language program using 8085 instruction set
3. Developing assembly language program using 8086 instruction set
4. Developing assembly language program using 8051 instruction set
5. Developing various I/O programs for 9085, 8086 and 8051

Unit: I

Intel 8085 Microprocessor: Introduction - Need for Microprocessors – Evolution – Intel 8085 Hardware - Architecture – Pin description - Internal Registers – Arithmetic and Logic Unit – Control Unit – Instruction word size - Addressing modes – Instruction Set – Assembly Language Programming - Stacks and Subroutines - Timing Diagrams. Evolution of Microprocessors – 16-bit and 32-bit microprocessors.

Unit II

Intel 8085 Interrupts and DMA: 8085 Interrupts – Software and Hardware Interrupts – 8259 Programmable Interrupt Controller - Data Transfer Techniques – Synchronous, Asynchronous and Direct Memory Access (DMA) and 8237 DMA Controller- 8253 Programmable Interval Timer.

Unit III

Memory & I/O Interfacing: Types of memory – Memory mapping and addressing – Concept of I/O map – types – I/O decode logic – Interfacing key switches and LEDs – 8279 Keyboard/Display Interface - 8255 Programmable Peripheral Interface – Concept of Serial Communication – 8251 USART – RS232C Interface.

Unit IV

Intel 8086 Microprocessor: Introduction-Intel 8086 Hardware – Pin description – External memory Addressing – Bus cycles – Interrupt Processing. Addressing modes - Instruction set – Assembler Directives.

Unit V

Microcontroller: Intel 8051 Microcontroller: Introduction – Architecture – Memory Organization – Special Function Registers – Pins and Signals – Timing and control – Port Operation – Memory and I/O interfacing – Interrupts – Instruction Set and Programming.

(Total : 60 Periods)

Content beyond Syllabus:

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,
PUDUCHERRY - 605 107

REMOTE BASED HEALTH MONITORING USING NODEMCU ESP8266

PROJECT REPORT

Submitted by

J.KISHOR KUMAR — REGISTER NO: 17TH1228

M.MADHAN KAVI — REGISTER. NO: 17TH1231

T.SUNDARAVARATHAN — REGISTER. NO: 17TH1248

Under the Guidance of

Dr. A.MEIAPPANE M.TECH., Ph. D

Associate Professor

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107

PONDICHERRY UNIVERSITY

INDIA

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

NodeMCU esp8266

4GB ROM

DC Power supply

Pulse Sensor

BP Sensor

Temperature Sensor

Motion Sensor

Smart Phone

3.2 SOFTWARE REQUIREMENTS

Firebase

Blynk

Android application

3.3ABOUT THE HARDWARE

NODEMCU ESP8266

The NodeMCU (*N*ode *M*icro*C*ontroller *U*nit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for the Internet of Things (IoT) projects of all kinds.

However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the "computer" on the chip. You also have to program it in low-level machine

ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply. Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

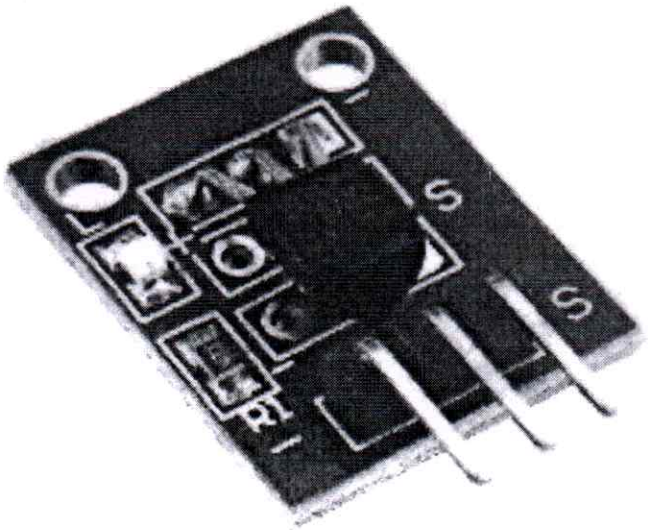


Fig.3.2.3 DS18B20 digital thermometer

BLOOD PRESSURE SENSOR

We are utilizing an arm type of electronic circulatory strain screen that not just has the ability give circulatory strain yet in addition measure beat rate. A computerized circulatory strain machi utilizes Oscillometric estimations and electronic computations. They are simple operable and g practically the right qualities. They can quantify systolic and diastolic pressing factors.

IT-T45 JAVA PROGRAMMING

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T45	Java Programming	3	1	---

Pre-requisite: Object Oriented Programming

Course Objectives:

1. To understand the basics of Java
2. To learn the features of Java
3. To learn the advanced concepts in Java.

Course Outcomes: Students will understand the benefits and capabilities of Java.

Unit: I

Creation of Java, importance of Java to internet, byte code, Java buzzwords, data types, declaring variables, dynamic initialization, scope and life time of variables, arrays, operators, control statements, type conversion and casting, compiling and running of simple Java program. Concepts of classes and objects, class fundamentals Declaring objects, assigning object reference variables, introducing methods, constructors, usage of static with data and methods, usage of final with data, access control, this key word, garbage collection, overloading methods and constructors, parameter passing - call by value, recursion, nested classes and inner classes, exploring the String class.

Unit: II

Basic concepts, member access rules, usage of super key word, forms of inheritance, method overriding, abstract classes, dynamic method dispatch, using final with inheritance, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.

Unit: III

Concepts of Exception handling, types of exceptions, usage of try, catch, throw, throws and finally keywords, Built-in exceptions, creating own exception sub classes, Concepts of Multithreading, differences between process and thread, thread life cycle ,creating multiple threads using Thread class, Runnable interface, Synchronization, thread priorities, inter thread communication, daemon threads, deadlocks, thread groups.

Unit: IV

Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes.

AWT : Concepts of components, container, panel, window, frame, canvas, Font class, Color class and Graphics.

Applets - Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.

Unit: V

RMI- JDBC- Developing Java Program for RMI and JDBC.

(Total : 60 Periods)

Content beyond Syllabus:

Java's support in graphics, networking, web technology etc.

Text Books:

1. The Complete Reference Java J2SE 5th Edition, Herbert Schildt, TMH Publishing Company Ltd, NewDelhi.
2. Big Java 2nd Edition, Cay Horstmann, John Wiley and Sons.

Reference Books:

1. Java How to Program, Sixth Edition, H.M.Dietel and P.J.Dietel, Pearson Education/PHI
2. Core Java 2, Vol 1, Fundamentals, Cay.S.Horstmann and Gary Cornell, Seventh Edition, Pearson Education.
3. Core Java 2, Vol 2, Advanced Features, Cay.S.Horstmann and Gary Cornell, Seventh Edition, Pearson Education.

Websites:

1. http://www.ibm.com/developerworks/java/
2. http://docs.oracle.com/javase/tutorial/rmi/.
3. IBM's tutorials on Swings, AWT controls and JDBC.

IT-T54 DATABASE MANAGEMENT SYSTEMS

Subject Code	Subject Name	Lectures (Periods)	Tutorials (Periods)	Practical (Periods)
IT-T54	Data Base Management Systems	3	1	0

Pre-requisite:

1. Knowledge about Memory Management
2. Programming Skills

Course Objectives:

To introduce the fundamental concepts of Database Management System to the students and to make them understand the usage of Database Management System in the current industry scenario.

Course Outcomes:

The students can be able to understand the concepts of Database Management System and to choose and design the database for the specific requirement of the project.

Unit I

Introduction: Introduction to Database Systems: Overview – Data Models – Database System Architecture – History of Database Systems. Entity-Relationship Model: Basic Concepts – Constraints – Keys – Design Issues – Entity Relationship Diagram – Weak Entity Sets – Extended E-R Features – Design of an E-R Database Schema.

Unit II

Relational Model: Structure of Relational Databases – Relational Algebra – Extended - Relational Algebra Operations – Modification of Database – Views – Tuple Relational - Calculus – Domain Relational Calculus. SQL: Background – Basic Structure – Set - Operations – Aggregate Functions – Null Values – Nested Sub-queries – Views – Complex Queries – Modification of the database –Joined Relations – Data-Definition Language.

Unit III

Integrity and Security: Domain Constraints – Referential Integrity – Assertions –Triggers – Security and Authorization – Authorization in SQL .Relational-Database Design: Normalization -First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form.

Unit IV

Storage and File Structures: Overview of Physical Storage Media – Magnetic Disks – RAID – Tertiary Storage – Storage Access – File Organization. Indexing and Hashing: Basic Concepts –Static Hashing – Dynamic Hashing.

Unit V

Transactions: Transaction concept – Transaction State – Implementation of Atomicity and Durability – Concurrent Executions – Serializability – Testing for Serializability. Concurrency Control: Lock-Based Protocols – Timestamp-Based Protocols. Recovery System: Failure Classification – Storage Structure – Recovery and Atomicity – Log-Based Recovery – Shadow Paging.

(Total : 60 Periods)

Content beyond Syllabus: The recent developments of the Database Management System and the current standards of the IT organizations have to be introduced to the students.

Text Books:

1. Silberschatz, Korth, Sudarshan, *Database System Concepts*, 6th Edition – McGraw-Hill Higher Education, International Edition, 2011.

Reference Books:

1. Fred R McFadden, Jeffery A Hoffer, Mary B. Prescott, Modern Database Management, Seventh Edition, Addison Wesley, 2004.
2. Elmasri, Navathe, Fundamentals of database Systems, Sixth Edition, Addison Wesley, 2010.
3. JefreyD.Ulman, Jenifer Widom, A First Course in Database Systems, Pearson Education Asia, 2001.
4. Bipin C Desai, An Introduction to Database Systems, Galgotia Publications Pvt Limited, 2003.

Websites:

1. http://www.database.com/
2. www.infoworld.com/t/dbms

PRINCIPAL
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY,
KALITHEERTHALKUPPAM,

SPAM EMAIL DETECTION WITH PROBABILISTIC DATA STRUCTURE USING JAVA

PROJECT REPORT

Submitted by

D.GOWRI **Reg.No:16TH1224**

Under the Guidance of

Dr. P.SIVAKUMAR M.E., Ph.D
Professor & Head

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF INFORMATION TECHNOLOGY

MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY
KALITHEERTHALKUPPAM, PUDUCHERRY-605 107.
PONDICHERRY UNIVERSITY
INDIA

MAY – 2021

1. The Java Programming Language

2. The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- High performance
- Architecture neutral
- Robust
- Object oriented
- Interpreted
- Portable
- Distributed
- Multithreaded
- Secure
- Dynamic

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

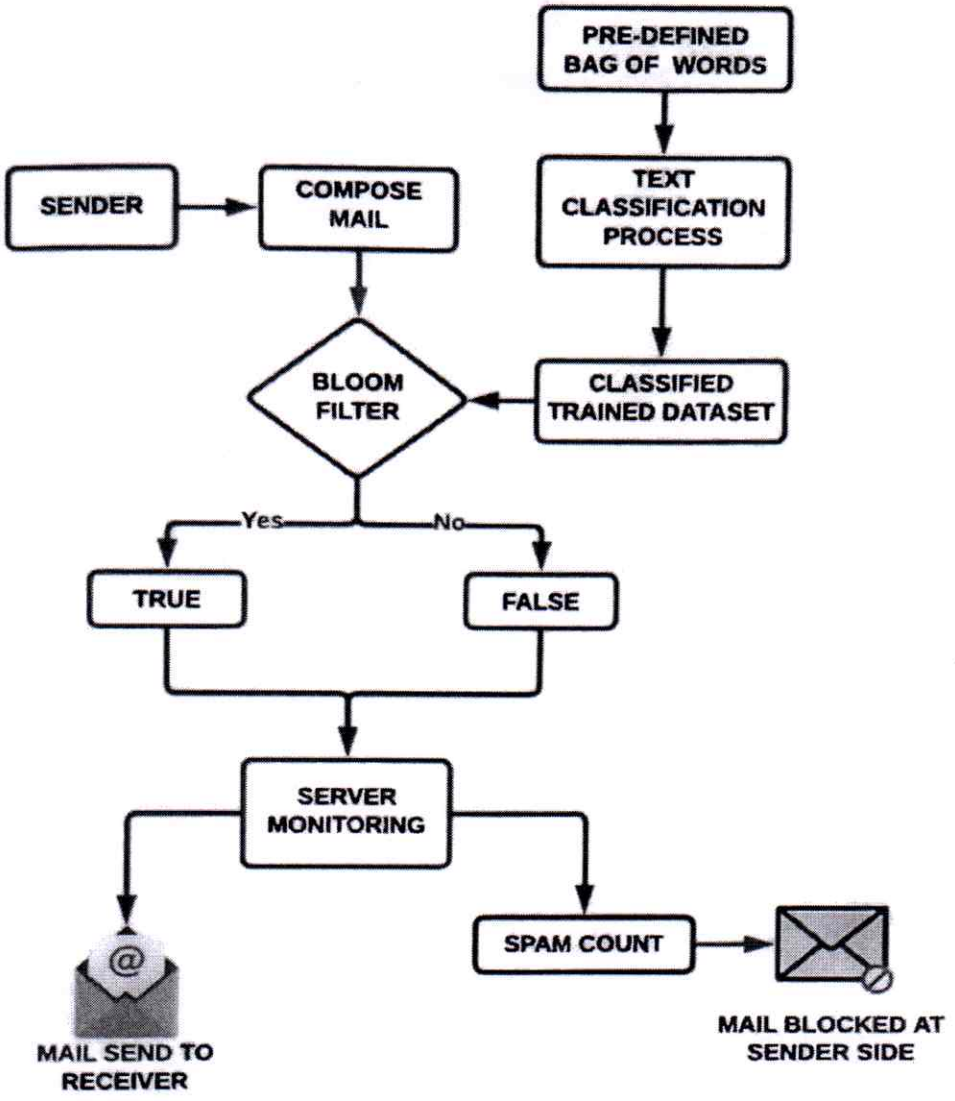


Figure 5.1 Proposed System Block Diagram

5.3 ALGORITHM USED IN PROPOSED WORK

In this module the bloom filter first checks the subject and body texts, there is any wrong word or unrelated special symbols. If it is presented then block mail and increase count. Otherwise bloom filter get the stream contents from the attachments and generate hash value for all streams.

- **Bloom Filter Techniques**

A Bloom filter is a space-efficient probabilistic data structure, conceived by Burton Howard Bloom in 1970, that is used to test whether an element is a member of a set. False positive matches are possible, but false negatives are not – in other words, a query returns

either "possibly in set" or "definitely not in set". Elements can be added to the set, but no more items added, the larger the probability of false positives.

An empty Bloom filter is a bit array of m bits, all set to 0. There must also be k different hash functions defined each of which maps or hashes some set element to one of the m array positions, generating a uniform random distribution. Typically, k is a small constant which depends on the desired false error rate ε, while m is proportional to k and the number of elements to be added. To add an element, feed it to each of the k hash functions to get k array positions. Set the bits at all these positions to 1.

To query for an element (test whether it is in the set), feed it to each of the k hash functions to get k array positions. If any of the bits at these positions is 0, the element is definitely not in the set; if it were, then all the bits would have been set to 1 when it was inserted. If all are 1, then either the element is in the set, or the bits have by chance been set to 1 during the insertion of other elements, resulting in a false positive

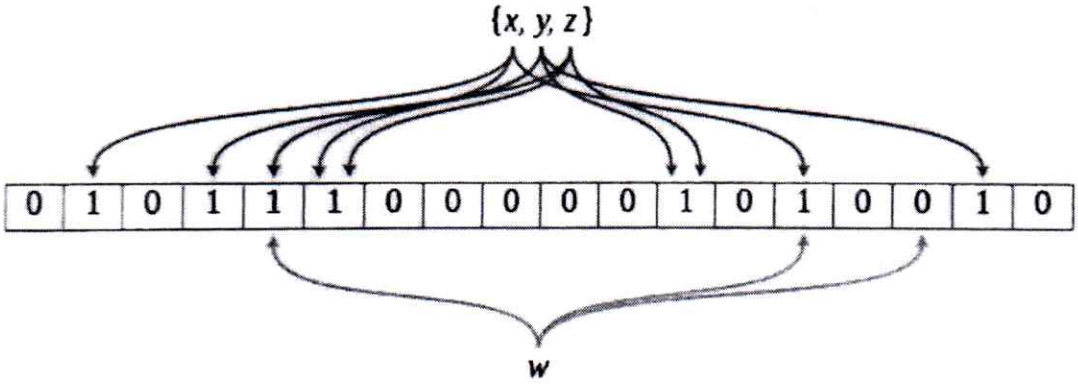


Fig 5.2 Bloom Filter Hash table structure

The spam filtering techniques are going to deploy on the sender side itself. By this the spam message cannot send to the receiver side. Spam email may also include malware as scripts or other executable file attachments. Before sending the mail it can able to filter the spam, normally some files come with an extension of .exe and encrypted file sent to the mails. Using filtering technique the junk mail can neglect the encrypted spam too. This system will improve bandwidth and memory storage. There are two techniques which is used to find the encrypted format text, Word Net dictionary and short message technique. The Bloom filters are used to find the junk mail, it have an advantage on other data structures for representing sets, such as self-balancing binary search tree. This prevents text based spam

filters from detecting and blocking spam messages. And it is used to improve the performance of Online Social Network. A Bayesian filter has a list of keywords along with their probabilities to identify an email as a spam email or a legitimate email. The spamming bot should deliver spam messages to a wide range of unique REAs for efficient spam delivery.

➢ Algorithm of Bloom Filter

❖ Algorithm 1 ABF: Algorithm for Insertion
Require: B, the bit-vector and v, input element
Ensure: N, number of additional hash function

1: if all $B[H1..k(v)] = 1$ then
2: $N \Leftarrow 1$
3: while $B[Hk+N(v)] = 1$ do
4: $N \Leftarrow N + 1$
5: end while
6: $B[Hk+N(v)] = 1$
7: else
8: all $B[H1..k(v)] = 1$
9: end if

❖ Algorithm 2 ABF: Algorithm for a Query
Require: B, the bit-vector and v, input element
Ensure: N, number of additional hash function

1: if all $B[H1..k(v)] = 1$ then
2: $N \Leftarrow 1$
3: while $B[Hk+N(v)] = 1$ do
4: $N \Leftarrow N + 1$
5: end while
6: return N
7: end if

➢ Pseudocode

```
for(int i=0;i<arr.length;i++)
     {
  Matcher regexMatcher = regex.matcher((arr[i]));
if(regexMatcher.find() || arr[i].length()>45)
{
    ++count;
if(count==0)
encrypt=arr[i] ;
```