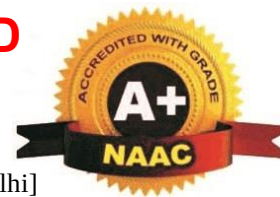




VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN

(An Autonomous Institution)

[Sponsored by Lavu Educational Society, Affiliated to JNTUH & Approved by AICTE, New Delhi]
Kondapur (V), Ghatkesar (M), Medchal - Malkajgiri (D) - 501 301. Phone: 96529 10002/3



VR25ITCS

VMTW

B.Tech.in INFORMATION TECHNOLOGY COURSE STRUCTURE – VR25

II B. Tech – I Semester

S. No	Course Code	Course	L	T	P	Credits
1	25MA301BS	Mathematical and Statistical Foundations	3	0	0	3
2	25CS302PC	Java Programming	3	0	0	3
3	25EC303PC	Computer Organization and Microprocessor	3	0	0	3
4	25CS304PC	Operating Systems	3	0	0	3
5	25IT305PC	Introduction to IoT	3	0	0	3
6	25MA306PC	Computational Mathematics Lab	0	0	2	1
7	25CS307PC	Java Programming Lab	0	0	2	1
8	25CS308PC	Operating Systems Lab	0	0	2	1
9	25IT309PC	Internet of Things Lab	0	0	2	1
10	25DS310SD	Data Visualization - R Programming/Power BI/ Tableau/Google Chart	0	0	2	1
		Total	15	0	10	20

Note: L-Theory

T-Tutorial

P-Practical

C-Credits

B. Tech.

L T P C

II Year I Sem

3 0 0 3

MATHEMATICAL AND STATISTICAL FOUNDATIONS**(Common for IT, CSE(AI ML), CSE(DS))****Pre-requisites:** Mathematics courses of first year of study. **Objectives:** To learn

- The Number Theory basic concepts useful for cryptography etc.
- The theory of Probability, and probability distributions of single random variables. 3. The sampling theory and testing of hypothesis and making inferences.
- The curve fitting, correlation and regression for the given data.

Course outcomes: After learning the contents of this paper the student must be able to

- Apply the number theory concepts to cryptography domain.
- Apply the concepts of probability and distributions to some case studies. 3. Correlate the material of one unit to the material in other units.
- Resolve the potential misconceptions and hazards in each topic of study. 5. Fit the curve, correlation and regression for the given data.

UNIT-I: Basics of Number Theory 10L

Greatest Common Divisors and Prime Factorization: Greatest common divisors – The Euclidean algorithm–The fundamental theorem of arithmetic– Factorization of integers and the Fermat numbers. Congruences : Introduction to congruences– Linear congruences.

UNIT-II: Random Variables and Probability Distributions 8L

Concept of a Random Variable Discrete Probability Distributions Continuous - Probability Distributions– Mean of a Random Variable– Variance of a Random Variable **Discrete Probability Distributions:** Binomial Distribution–Poisson distribution

UNIT-III: Continuous Distributions and Sampling 10L

Uniform Distribution– Normal Distribution– Areas under the Normal Curve– Applications of the Normal Distribution Normal Approximation to the Binomial Distributions.

Fundamental Sampling Distributions: Random Sampling– Some Important Statistics– Sampling Distributions– Sampling Distribution of Means– Central Limit Theorem.

UNIT-IV: Tests of Hypotheses(LargeandSmallSamples) 10L

Statistical Hypotheses: General Concepts Testing a Statistical Hypothesis. Single sample: Tests concerning a single mean. Two samples: Tests on two mean (Unknown for equal variance). One sample: Test on a single proportion. Two samples: Tests on two proportions. Two- sample tests concerning variances: F-distribution

UNIT-V: Applied Statistics 10L

Curve fitting by the method of least squares– Fitting of straight lines– Second degree parabolas and more general curves– Correlation and Regression– Rank correlation.

TEXT BOOKS:

1. Kenneth H. Rosen, Elementary Number Theory & its Applications, sixth edition, Addison Wesley, ISBN 978 0-321-50031-1.
2. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, Probability &

3. Statistics for Engineers & Scientists, 9thEd. Pearson Publishers. SC Gupta and VK Kapoor, Fundamentals of Mathematical Statistics, Khanna publications.

REFERENCE BOOKS:

1. T.T. Soong, Fundamentals of Probability and Statistics for Engineers, JohnWiley& Sons, Ltd, 2004.
2. Sheldon MRoss, Probability and statistics for Engineers and scientists, academicpress.
3. SC Gupta and VK Kapoor, Fundamentals of Mathematical statistics, Khanna publications.

B. Tech.

L T P C

II Year I Sem

3 0 0 3

JAVA PROGRAMMING**Course Outcomes:**

- Identify the model of Object-Oriented Programming: Abstract data types, Encapsulation, Inheritance and Polymorphism.
- Summarize the fundamental features like Interfaces, Exceptions and Collections. 3. Correlate the advantages of Multi-threading.
- Design interactive programs using Applets, AWT and Swings. 5. Develop real time applications using the features of Java.

UNIT-I**OBJECT ORIENTED THINKING**

Introduction, Need of object-oriented programming, principles of object-oriented languages, Applications of OOP, history of JAVA, Java Virtual Machine, Java features, Program structures, Installation of JDK.

Variables, Primitive data types, Identifiers- Naming Conventions, Keywords, Literals, Operators- Binary, Unary and Ternary, Expressions, Primitive Type conversion and casting, flow of control- branching, conditional, loops.

UNIT-II**CLASSES, INHERITANCE, POLYMORPHISM**

Classes and Objects- Classes, Objects, creating objects, methods, constructors- constructor overloading, cleaning up unused objects- Garbage collector, class variable and methods- static keyword, this keyword, arrays, Command line arguments, Nested Classes.

Strings: String, String Buffer, String Tokenizer.

Inheritance and Polymorphism- Types of Inheritance, deriving classes using extends keyword, super keyword, Polymorphism- Method Overloading, Method Overriding, final keyword, abstract classes.

UNIT-III: INTERFACES, PACKAGES, EXCEPTIONS

Interfaces: Interface, Extending Interface, Interface Vs Abstract classes.

Packages- Creating Packages, using Packages, Access protection, java I/O package.

Exceptions - Introduction, Exception handling techniques-try...catch, throw, throws, finally block, user defined Exception.

UNIT-IV: MULTI THREADING, COLLECTIONS

java.lang.Thread, the main Thread, creation of new Threads, Thread priority, multithreading- using isAlive() and join(), Synchronization, suspending and resuming Threads, Communication between Threads. Exploring java.io, Exploring java.util.

Collections: Overview of Collection Framework: Array List, Vector, Tree Set, Hash Map, Hash Table, Iterator, Comparator.

UNIT-V:APPLETS, AWT AND SWINGS

Applet class, Applet structure, an example of Applet program, Applet life cycle.

Event Handling- Introduction, Event Delegation Model, Java.awt.event Description, Adapter classes, Inner classes.

Abstract Window Toolkit: Introduction to AWT, components and containers, Button, Label, Checkbox, Radio buttons, List boxes, choice boxes, Text field and Text area, container classes, Layout Managers.

Swing: Introduction, JFrame, JApplet, JPanel, Components in Swings, JList and JScroll Pane, SplitPane, JTabbed Pane, Dialog Box, Pluggable Look and feel.

TEXT BOOKS:

1. Java: The Complete Reference, 10th edition, Herbert Schildt, McGraw Hill.
2. Java Fundamentals: A Comprehensive Introduction, Herbert Schildt and Dale Skrien, TMH.
3. Java for Programming, P.J.Dietel Pearson Education.

REFERENCE BOOKS:

1. Object Oriented Programming through Java, P.Radha Krishna, Universities Press.
2. Thinking in Java, Bruce Eckel, Pearson Education.
3. Programming in Java, S.Malhotra and S.Choudhary, Oxford University Press.

B. Tech.	L	T	P	C
II Year I Sem	3	0	0	3

COMPUTER ORGANIZATION AND MICROPROCESSOR

Course Objectives:

- To understand basic components of computers.
- To understand the architecture of the 8086 processor.
- To understand the instruction sets, instruction formats and various addressing modes of 8086.
- To understand the representation of data at the machine level and how computations are performed at machine level.
- To understand the memory organization and I/O organization.
- To understand the parallelism both in terms of single and multiple processors.

Course Outcomes:

- Able to understand the basic components and the design of CPU, ALU and Control Unit.
- Ability to understand memory hierarchy and its impact on computer cost/performance.
- Ability to understand the advantage of instruction level parallelism and pipelining for high performance Processor design.
- Ability to understand the instruction set, instruction formats and addressing modes of 8086.
- Ability to write assembly language programs to solve problems.

UNIT-I

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

Basic Computer Organization and Design: Instruction codes, Computer Registers, Computer instructions, Timing and Control, Instruction cycle, MemoryReference Instructions, Input– Output and Interrupt, Complete Computer Description.

Micro Programmed Control: Control memory, Address sequencing, micro program example, design of control unit.

UNIT-II

Central Processing Unit: The 8086 Process or Architecture, Register organization, Physical memory organization, General Bus Operation, I/O Addressing Capability, Special Processor Activities, Minimum and Maximum mode system and timings.

8086 Instruction Set and Assembler Directives- Machine language instruction formats, Addressing modes, Instruction set of 8086, Assembler directives and operators.

UNIT-III

Assembly Language Programming with 8086- Machine level programs, Machine coding the programs, Programming with an assembler, Assembly Language example programs.

Stack structure of 8086, Interrupts and Interrupt service routines, Interrupt cycle of 8086, Interrupt programming, Passing parameters to procedures, Macros, Timings and

Delays.

UNIT-IV

Computer Arithmetic: Introduction, Addition and Subtraction, Multiplication Algorithms, Division Algorithms, Floating-point Arithmetic operations. Input-Output Organization:

Peripheral Devices, Input- Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt, Direct memory Access, Input –Output Processor (IOP), Intel 8089 IOP.

UNIT-V

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors.

TEXT BOOKS:

1. Computer System Architecture, M.Morris Mano, Third Edition, Pearson.(UNITS– I,IV, V)
2. Advanced Microprocessors and Peripherals, K M Bhurchandi,A. K Ray ,3rd edition, McGraw Hill India Education Private Ltd. (UNITS - II, III).

REFERENCE BOOKS:

1. Micro processors and Interfacing, DV Hall, SSSP Rao, 3rd edition, McGraw Hill India Education Private Ltd.
2. Carl Hamacher, Zvonko Vranesic, SafwatZaky: Computer Organization, 5th Edition,Tata McGraw Hill.
3. Computer Organization and Architecture, William Stallings,9thEdition, Pearson.
4. David A. Patterson, John L. Hennessy: Computer Organization and Design– The Hardware / Software Interface ARM Edition,4th Edition, Elsevie

B. Tech.

L T P C

II Year I Sem

3 0 0 3

OPERATING SYSTEMS**Prerequisites:**

- A course on “Computer Programming and Data Structures”. 2. A course on “Computer Organization and Architecture”.

Course Objectives:

- Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection)
- Introduce the issues to be considered in the design and development of operating system
- Introduce basic Unix commands, system call interface for process management, inter process communication and I/O in Unix

Course Outcomes:

- Will be able to control access to a computer and the files that may be shared
- Demonstrate the knowledge of the components of computers and the irrespective roles in computing.
- Ability to recognize and resolve user problems with standard operating environments.
- Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

UNIT-I

Operating System - Introduction, Structures - Simple Batch, Multi programmed, Time- shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls

Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads

UNIT-II

CPU Scheduling – Scheduling Criteria, Scheduling Algorithms, Multiple - Processor Scheduling. System call interface for process management-fork, exit, wait, waitpid, exec **Deadlocks** - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

UNIT-III

Process Management and Synchronization-The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors

Inter process Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT-IV

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

UNIT-V

File System Interface and Operations-

Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

TEXT BOOKS:

1. Operating System Principles-Abraham Silber chatz, Peter B.Galvin, Greg Gagne 7th Edition, John Wiley
2. Advanced programming in the UNIX environment, W.R.Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems- Internals and Design Principles, William Stallings, Fifth Edition– 2005, Pearson Education/PHI
2. Operating System A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI
4. UNIX programming environment, Kernighan and Pike, PHI/Pearson Education
5. UNIX Internals- The New Frontiers, U.Vahalia, Pearson Education.

B. Tech.**L T P C****II Year I Sem****3 0 0 3**

INTRODUCTION TO IOT

Course Objectives:

- Understand the concepts of Internet of Things and able to build IoT applications
- Learn the programming and use of Arduino and Raspberry Pi boards.
- Know about data handling and analytics in SDN.

Course Outcomes: Upon completing this course, the student will be able to:

- Understand key characteristics, layers, protocols, sensor networks and their role in IoT.
- Differentiate between IoT and M2M communications and achieve interoperability using Arduino Programming.
- Apply Python constructs on Raspberry Pi interfacing models and their use cases.
- Design and implement SDN-based solutions for IoT deployments using RaspberryPi
- Apply the knowledge of cloud storage models to select an appropriate model for any IoT application.
- Analyze different sensor technologies for real world applications through IoT.

UNIT-I**Introduction to IoT and Sensor Networks**

Introduction to Internet of Things (IoT), Characteristics and Applications of IoT, IoT Architecture and Reference Models (IETF, ITU-T)

Physical Design of IoT: Devices, Gateways, and Data Centers

Functional Blocks of IoT: Sensing, Actuation, Communication, Enabling Technologies: RFID, Wireless Sensor Networks, Nanotechnology

Basics of Networking and Communication Protocols: MQTT, CoAP, ZigBee, HTTP

Sensor Networks: Types, Topologies, and Protocols Introduction to IoT Security and Privacy Fundamentals.

UNIT-II**Machine to Machine (M2M) and Embedded Programming for IoT**

Machine-to-Machine Communications Overview, Difference between IoT and M2M, Interoperability in IoT: Standards and Protocols.

Introduction to Arduino Programming for IoT, Integration of Sensors and Actuators with Arduino. Hands-on Exercises: Sensor Data Acquisition and Actuator Control, Basic Communication Protocols Implementation on Arduino, IoT Device Interoperability Challenges and Solutions.

UNIT-III**RaspberryPi with Python Programming for IoT**

Introduction to Python Programming Basics

Overview of RaspberryPi and Its Role in IoT, Interfacing RaspberryPi with Sensors and Actuators (UART, SPI, I2C).

Data Acquisition and Local Processing on RaspberryPi, Sending Data to Cloud Platforms. Implementation of IoT Projects Using Raspberry Pi

Case Studies: Smart Home Automation, Healthcare Monitoring, Environmental Sensing

UNIT-IV

Network Management and Software Defined Networking(SDN) for IoT

Introduction to Software Defined Networking(SDN), SDN Architecture: Controller, Data Plane, Control Plane, Benefits of SDN in IoT: Flexibility, Centralized Control, Security,

Traffic Management, SDN Models: Open SDN, API SDN, Overlay, Hybrid Data Handling, Analytics, and Security Considerations in SDN for IoT

Network Management Challenges in IoT, Integration of SDN with IoT Networks

UNIT-V

Cloud and Edge Computing Models with IoT Use Cases

Introduction to Cloud Computing and Cloud Storage Models, Edge and Fog Computing Concepts for IoT, Web Servers and Cloud Platforms for IoT (AWS IoT, Azure IoT, etc.), Sensor-Cloud Architecture Data Analytics and Visualization Techniques for IoT Data IoT Use Cases: Smart Cities, Smart Homes, Connected Vehicles, Smart Grid,

Industrial IoT Case Studies: Agriculture, Healthcare, Activity Monitoring.

TEXTBOOKS

1. Pethuru Raj and Anupama C.Raman "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press.
2. Vijay Madisetti, Arshdeep Bahga, "Internet of Things: A Hands-On Approach"

REFERENCEBOOKS

1. Terokarvinen, kemo, karvinen and villeyvaltokari, "Makesensors": 1st edition, M 2014, Maker media.
2. Waltenegus Dargie, Christian Poellabauer, Fundamentals of Wireless Sensor Networks: Theory and Practice.
3. Charles Bell, Beginning Sensor networks with Arduino and RaspberryPi, 2013, Apress.
4. Fei Hu, Security and Privacy in Internet of Things(IoTs), CRC Press, Taylor & Francis Group.
5. S. Sahoo, S. Sahoo, S. Mishra, Software-Defined Networking for Future Internet Technology: Concepts and Applications, Routledge.
6. Honbo Zhou, The Internet of Things in the Cloud: A Middleware Perspective. 2012, CRC Press (Taylor & Francis).

B. Tech.

L T P C

II Year I Sem

0 0 2 1

COMPUTATIONAL MATHEMATICS LAB**(Using Python/MATLAB software)****Pre-requisites:** Matrices, Iterative methods and ordinary differential equations**Course Objectives:** To learn

- Solve problems of Eigen values and Eigen Vectors using Python/MATLAB.
- Solution of Algebraic and Transcendental Equations using Python/MATLAB.
- Solve problems of Linear system of equations
- Solve problems of **First-Order ODEs Higher order linear differential equations with** constant coefficients

Course outcomes: After learning the contents of this paper, the student must be able to

- Develop the code to find the Eigen values and Eigen Vectors using Python/MATLAB.
- Develop the code find solution of Algebraic and Transcendental Equations and Linear system of equations using Python/MATLAB
- Write the code to solve problems of **First-Order ODEs Higher order linear differential equations with** constant coefficients

Visualize all solutions Graphically through programmes*UNIT-I: Eigen values and Eigen vectors:****6P****Programs:**

- Finding real and complex Eigen values.
- Finding Eigen vectors.

UNIT-II: Solution of Algebraic and Transcendental Equations 6P

Bisection method, Newton Raphson Method

Programs:

- Root of a given equation using Bisection method.
- Root of a given equation Newton Raphson Method.

UNIT-III: Linear system of equations: 6P

Jacobi's iteration method and Gauss-Seidaliteration method

Programs:

- Solution of given system of linear equations using Jacobi's method
- Solution of given system of linear equations using Gauss-Seidal method

UNIT-IV: First-Order ODEs 8P

Exact and non-exact equations, Applications: exponential growth/decay, Newton's law of cooling.

Programs:

- Solving exact and non-exact equations
- Solving exponential growth/decay and Newton's law of cooling problems

UNIT-V: Higher order linear differential equations with constant coefficient**6P****Programs:**

- Solving homogeneous ODEs
- Solving non-homogeneous ODEs

TEXT BOOKS:

1. MATLAB and its Applications in Engineering, Rajkumar Basal, Ashok Kumar Geo, Manoj Kumar Sharma, Pearson publication.
2. Kenneth A.Lambert, The fundamentals of Python: First Programs, 2011, Cengage Learnings.
3. Think Python First Edition, by Allen B.Downey, Orielly publishing.
4. Introduction to Python Programming, William Mitchell, Povel Solin, MartinNovak et al., NCLab Public Computing, 2012.
5. Introduction to Python Programming, ©Jacob Fredslund,2007.

REFERENCE BOOKS:

1. An Introduction to Python, John C.Lusth, The University of Alabama,2011. 2. Introduction to Python, ©Dave Kuhlman, 2008.

B. Tech.

II Year I Sem

L	T	P	C
0	0	2	1

JAVA PROGRAMMING LAB**Course Outcomes:**

- Analyze a problem, identify and define the computing requirements appropriate to its solution
- Using object-oriented programming concepts.
- Design the applications using Inheritance, Polymorphism and Synchronization concepts.
- Handle exceptions at Compile time and Run time.
- Solve the real-world problems using Java Collection framework. 6. Develop GUI applications using Applets, AWT and Swings.

TASK1

Write java programs that implement the following

- Class and object
- Constructor
- Parameterized constructor
- d. Method overloading
- Constructor overloading.

TASK2

- Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.
- Write a Java program for sorting a given list of names in ascending order.
- Write a Java Program that reads a line of integers, and then displays each integer and the sum of all the integers (Use String Tokenizer class of java.util)

TASK3

Write java programs that uses the following key words

- this
- b) super
- c)static
- d)final

TASK4

- Write a java program to implement method overriding
- Write a java program to implement dynamic method dispatch.
- Write a Java program to implement multiple inheritance.
- Write a java program that uses access specifiers.

TASK5

- Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
- Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
- Write a Java program that displays the number of characters, lines and words in a text file

TASK6

- Write a Java program for handling Checked Exceptions.
- Write a Java program for handling Unchecked Exceptions.

TASK7

- Write a Java program that creates three threads. First thread displays "Good

Morning”every one second, the second thread displays“Hello”every two seconds and the third thread displays “Welcome” every three seconds.

- b) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.

TASK8

Write a program illustrating following collections framework

- a) Array List b)Vector c)Hash Table d)Stack

TASK9

- a) Develop an applet that displays a simple message.
- b) Develop an applet that receives an integer in one text field and compute its factorial value and return it in another text field, when the button named “Compute” is clicked.
- c) Write a Java program that works as a simple calculator. Use a grid layout to arrange button for the digits and for the +, -,*, % operations. Add a text field to display the result.

TASK10

- a) Write a Java program for handling mouse events.
- b) Write a Java program for handling key events.

TASK11

- a) Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields Num1 and Num 2.
- b) The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1or Num2 were not an integer, the program would throw Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception and display the exception in a message dialog box.

TASK12

- a) Write a java program that simulates traffic light. The program lets the user select one of three lights: red, yellow or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts
- b) Write a Java program that allows the user to draw lines, rectangles and ovals.
- c) Create a table in Table.txt file such that the first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using JTable component.

TEXT BOOKS:

1. Java: The Complete Reference, 10 edition, Herbert Schildt, McGraw Hill.
2. Java Fundamentals-A Comprehensive introduction, Herbert Schildt and Dale krien, TMH.
3. Java for programming, P.J. Dietel Pearson education (OR) Java: How to Program P.J. Dietel and H.M. Dietel, PHI

REFERENCE BOOKS:

1. Object Oriented Programming through java, P. Radha Krishna, Universities Press.
2. Thinking in Java, Bruce Eckel, Pearson Education
3. Programming in Java, S. Malhotra and S. Choudhary, Oxford University Press.S. Choudhary,OxfordUniversityPress.

B. Tech.

L T P C

II Year I Sem

0 0 2 1

OPERATING SYSTEMS LAB**Prerequisites:**

- A course on “Programming for Problem Solving”.
- A course on “Computer Organization and Architecture”. **Co-requisite:** A course on “Operating Systems”.

Course Objectives:

- To provide an understanding of the design aspects of operating system concepts through simulation.
- Introduce basic Unix commands, system call interface for process management, inter process communication and I/O in Unix.

Course Outcomes:

- Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
- Able to implement C programs using Unix system calls.

List of Experiments:

1. Write C programs to simulate the following CPU Scheduling algorithms
a)FCFS b)SJF c) Round Robin d) priority
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, lseek, stat, fork, exit)
3. Write a C program to simulate Bankers Algorithm for Dead lock Avoidance.
4. Write a C program to implement the Producer– Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write C programs to illustrate the following IPC mechanisms
a) Pipes b) FIFOs c) Message Queues d) Shared Memory
6. Write C programs to simulate the following memory management techniques
a)Paging b) Segmentation
7. Write C programs to simulate Page replacement policies
1. FCFS b) LRU c) Optimal

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Advanced programming in the Unix environment, W. R. Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems–Internals and Design Principles, William Stallings, Fifth Edition–2005, Pearson Education/PHI.
2. Operating System-A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S Tanenbaum, 2nd edition, Pearson/PHI.
4. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education.
5. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education.

B. Tech.**L T P C****II Year I Sem****0 0 2 1****INTRODUCTION TO IOT LAB****Course Objectives:**

- To introduce the raspberry PI platform, that is widely used in IoT applications
- To introduce the implementation of distance sensor on IoT devices

Course Outcomes:

- Illustrate various IoT devices, sensors, and actuators, including Arduino, Raspberry Pi, and micro controllers.
- Identify sensors technologies used in IoT applications and their working principles.
- Apply programming languages like Python, C/C++ in IoT applications.
- Develop skills in collecting data from sensors and processing it using micro controllers or edge computing devices.

List of Programs

1. Installing OS on Raspberry Pi
 - a. Installation using Pi Imager
 - b. Installation using image file
 - Downloading an Image
 - Writing the image to an SD card using Linux using Windows
 - Booting up Follow the instructions given in the URL
<https://www.raspberrypi.com/documentation/computers/gettingstarted.html>
2. Accessing GPIO pins using Python
 - a. Installing GPIO Zero library.
First, update your repositories list: Sudo apt update
Then install the package for Python3: Sudo apt install python3-gpiozero
 - b. Blinking an LED connected to one of the GPIO pin
 - c. Adjusting the brightness of an LED Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) using the in-built PWM wavelength.
3. Using Raspberry Pi
 - a. Calculate the distance using a distance sensor.
 - b. Basic LED functionality using switch.
4. Using Arduino
 - a. Calculate the distance using a distance sensor.
 - b. Basic LED functionality using switch.
 - c. Calculate temperature using a temperature sensor
5. Using Node MCU
 - a. Including required libraries of Node MCU
 - b. Calculate the distance using a distance sensor.
 - c. Basic LED functionality using switch.
 - d. Calculate temperature using a temperature sensor
6. Using ESP32
 - a. Including required libraries of ESP32.
 - b. Calculate the distance using a distance sensor.
 - c. Basic LED functionality using switch.
 - d. Calculate temperature using a temperature sensor.

7. Collecting Sensor Data

a. DHT Sensor interface

- Connect the terminals of DHT GPIO pins of Raspberry Pi.
- Import the DHT library using `import Adafruit_DHT`.
- Read sensor data and display it on screen.
- Read sensor data and display it on the Cloud platform using ThingSpeak (API).
- Read sensor data and display it through the Blynk App.

8. Capstone Project: End-to-End IoT Solution Integrating Sensors, Network, and Cloud.

TEXT BOOKS:

1. Arshdeep Bahga and Vijay Madisetti, Internet of Things-A Hands-on Approach, 2015, Universities Press.
2. Matt Richardson & Shawn Wallace, Getting Started with Raspberry Pi, 2014, O'Reilly.

REFERENCE BOOKS:

1. Bernd Scholz-Reiter, Florian Michahelles, "Architecting the Internet of Things", ISBN978-3-642-19156-5 e-I SBN 978-3-642-19157-2, Springer, 2016.
2. N. Ida, Sensors, Actuators and Their Interfaces, Scitech Publishers, 2014.

B. Tech.	L	T	P	C
II Year I Sem	0	0	2	1

**DATA VISUALIZATION-RPROGRAMMING/ POWERBI/TABLEAU/
GOOGLECHART**

Course Objectives:

- Effective use of Business Intelligence (BI)technology(Tableau) to apply data visualization
- To discern patterns and relationships in the data. 3. To build Dashboard applications.
- To communicate the results clearly and concisely.
- To be able to work with different formats of datasets.

Course Outcomes:

- Understand how to import data into Tableau.
- Understand concepts of Dimensions and Measures.
- Develop Programs and understand how to map Visual Layouts and Graphical Properties
- Create Dashboard, custom charts, and, publish to tableau online for any real time dataset

List of Programs

1. Understanding Data, what is data, where to find data, Foundations for building Data Visualizations, Creating Your First visualization.
2. Getting started with Tableau Software using Data file formats, connecting your Data to Tableau, creating basic charts (line, bar charts, Tree maps), Using the Show me panel.
3. Tableau Calculations, Overview of SUM ,AVR, and Aggregate features, Creating custom calculations and fields.
4. Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Tools and Menus, Formatting specific parts of the view.
5. Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data
6. Structuring your data, Sorting and filtering Tableau data, Pivoting Tableau data.
7. Advanced Visualization Tools: Using Filters, Using the Detail panel, using the Size panels, customizing filters, Using and Customizing tooltips, Formatting your data with colors.
8. Creating Dashboards & Storytelling, creating your first dashboard and Story, Design for different displays, adding interactivity to your Dashboard, Distributing & Publishing your Visualization.
9. Tableau file types, publishing to Tableau Online, sharing your visualizations, printing, and exporting.
10. Creating custom charts, cyclical data and circular area charts, Dual Axis charts.
11. Visualize various data patterns taking any dataset fromKaggle.
12. Visualize data patterns using Google Charts by creating interactive Line, Bar, and Pie charts along with Dashboards using HTML and JavaScript.

REFERENCEBOOKS:

1. Brett Powell, Microsoft Power BI cook book, 2ndedition.
2. Roger D. Peng, R Programming for Data Science
3. Norman Matloff Cengage Learning India, The Art of R Programming.

WEBRESOURCE:

1. <https://developers.google.com/chart/interactive/docs>.