

VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN
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

II B. Tech. - II Semester

S.No	Course Code	Course	L	T	P	Credits
1	25CS401PC	Discrete Mathematics	3	0	0	3
2	25CS402PC	Data Communications and Computer Networks	3	0	0	3
3	25CS403PC	Database Management Systems	3	0	0	3
4	25CS404PC	Formal Languages and Automata Theory	3	0	0	3
5	25IT405PC	Web Programming	3	0	0	3
6	25MS406BS	Innovation and Entrepreneurship	2	0	0	2
7	25CS407PC	Database Management Systems Lab	0	0	2	1
8	25CS408PC	Computer Networks lab	0	0	2	1
9	25IT409PC	Web Programming Lab	0	0	2	1
10	25ML410SD	Node JS/ React JS/ Django, UI Design - Flutter	0	0	2	1
11	25MS411BS	Indian Knowledge System	1	0	0	1
		Total	18	0	08	22

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(AIML), 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 irrelative 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 an 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 7thEdition, 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

Raspberry Pi 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 Madiseti, Arshdeep Bahga, "Internet of Things: A Hands-On Approach"

REFERENCEBOOKS

1. Terokarvinen, kemo, karvinen and villeyvaltokari, "Makesensors": 1st edition, M 2014, Maker media.
2. Walteneus 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.	L	T	P	C
II Year I Sem	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, 2ndedition, 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 Thing Speak (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 Axisc harts.
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>.

B. Tech.

L T P C

II Year II Sem

3 0 0 3

DISCRETE MATHEMATICS**Course Objectives:**

- Introduces elementary discrete mathematics for computer science and engineering.
- Topics include formal logic notation, methods of proof, induction, sets, relations, algebraic structures, elementary graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.

Course Outcomes:

- Understand and construct precise mathematical proofs
- Apply logic and set theory to formulate precise statements
- Analyze and solve counting problems on finite and discrete structures
- Describe and manipulate sequences
- Apply graph theory in solving computing problems

UNIT-I

Mathematical logic: Introduction, Statements and Notation, Connectives, Normal Forms, Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

UNIT-II

Set theory: Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures, Relations and Ordering, Functions.

UNIT-III

Algebraic Structures: Introduction, Algebraic Systems, Semi groups and Monoids, Lattices as Partially Ordered Sets, Boolean Algebra.

UNIT-IV

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutation with Constrained Repetitions, Binomial Coefficient, The Binomial and Multinomial Theorems, The Principle of Exclusion.

UNIT-V

Graph Theory: Basic Concepts, Isomorphism and Sub graphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color Problem.

TEXTBOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R. Manohar, McGraw-Hill, 1st ed.
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joel Mott, Abraham Kandel, Theodore P. Baker, Prentis Hall of India, 2nd ed.

REFERENCEBOOKS:

1. Discrete and Combinatorial Mathematics – an applied introduction: Ralph. P. Grimald, Pearson education, 5th edition.
2. Discrete Mathematical Structures: Thomas Kosy, Tata Mc Graw Hill Publishing co.

B. Tech.

L T P C

II Year II Sem

3 0 0 3

DATA COMMUNICATIONS AND COMPUTER NETWORKS**Course Objectives:**

- Explore basic concepts of data communication
- Introduce students to TCP/IP and OSI models along with their merits and demerits.
- Explore in detail services offered by various layers of OSI Model.
- Understand link layer protocols, UDP, TCP and application layer protocols.

Course Outcomes:

- Understand and explore the basics of communication and computer networks
- Understand data link, network and transport layers concepts of a computer network.
- Understand the working of application layer protocols.

UNIT-I

Data Communications: Components Direction of Data flow Networks Components and Categories— Types of Connections Topologies—Protocols and Standards— ISO/ OSI model, Example Networks such as ATM, Frame Relay, ISDN
Physical layer: Transmission modes, Multiplexing, Transmission Media, Switching, Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks.

UNIT-II

Data link layer: Introduction, Framing, and Error— Detection and Correction— Parity— LRC— CRC Hamming code, Flow and Error Control, Noiseless Channels, Noisy Channels, HDLC, Point to Point Protocols.111 Medium Access sub layer: ALOHA, CSMA/CD,LAN—EthernetIEEE802.3,IEEE802.5 – IEEE 802.11, Random access, Controlled access, Channelization.

UNIT-III

Network layer: Logical Addressing, Internetworking, Tunneling, Address mapping, ICMP, IGMP, Forwarding, Uni-Cast Routing Protocols, Multicast Routing Protocols.

UNIT-IV

Transport Layer: Process to Process Delivery, UDP and TCP protocols, Data Traffic, Congestion, Congestion Control, QoS, Integrated Services, Differentiated Services, QoS in Switched Networks.

UNIT-V

Application Layer: Domain name space, DNS in Internet, Electronic Mail, SMTP, FTP, WWW, HTTP, SNMP.

TEXTBOOKS:

1. Data Communications and Networking, Behrouz A. Forouzan, Fourth Edition TMH.

REFERENCEBOOKS:

1. Computer Networks, Andrew S Tanenbaum, 6th Edition. Pearson Education.
2. Computer Networking: A Top-Down Approach Featuring the Internet. James F.Kurose & Keith W. Ross, 3rd Edition, Pearson Education
3. Data communications and Computer Networks, P. C Gupta, PHI.
4. An Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education.

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

DATABASE MANAGEMENT SYSTEMS

Prerequisites: A course on “Data Structures”.

Course Objectives:

- To understand the basic concepts and the applications of database systems. 2. To master the basics of SQL and construct queries using SQL.
- Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.
- **Course Outcomes:**
- Gain knowledge of fundamentals of DBMS, database design and normal forms 2. Master the basics of SQL for retrieval and management of data.
- Be acquainted with the basics of transaction processing and concurrency control. 4. Familiarity with database storage structures and access techniques

UNIT-I

Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS **Introduction to Database Design:** Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT-II

Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views. Relational Algebra, Tuple relational Calculus, Domain relational calculus.

UNIT-III

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multivalued dependencies, FOURTH normal form, FIFTH normal form.

UNIT-IV

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, Lock Based Protocols, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity, Log-Based Recovery, Recovery with Concurrent Transactions.

UNIT-V

Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and

Secondary Indexes, Index data Structures, Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes- Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index Structure.

TEXTBOOKS:

1. Database System Concepts, Silberschatz, Korth, Mc Graw hill, V edition.3rd Edition
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill

REFERENCEBOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elma sri Navrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S. Shahand V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
Fundamentals of Database Management Systems, M. L.Gillenson, Wiley Student Edition

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

FORMAL LANGUAGES AND AUTOMATA THEORY

Pre requisites: Mathematical Foundations Course Objectives

- To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
- To introduce the fundamental concepts of formal languages, grammars and automata theory. 3. Classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing.
- To understand deterministic and non-deterministic machines.
- To understand the differences between decidability and undecidability.

Course Outcomes

- Understand the concept of abstract machines and their power to recognize the languages. 2. Employ finite state machines for modeling and solving computing problems.
- Design context free grammars for formal languages. 4. Distinguish between decidability and undecidability.

UNIT-I

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory–Alphabets, Strings, Languages, Problems. **Nondeterministic Finite Automata:** Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with ϵ -transitions to NFA without ϵ -transitions. Conversion of NFA to DFA, Moore and Melay machines

UNIT-II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages, Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT-III

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Trees, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

Push Down Automata: Definition of the Push down Automata on, the Languages of a PDA,

Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.

UNIT-IV

Normal Forms for Context-Free Grammars: Eliminating useless symbols, Eliminating ϵ -Productions. Chomsky Normal form Greibach Normal form.

Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications **Closure Properties of Context-Free Languages:** Closure properties of CFL's, Decision Properties of CFL's Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

UNIT-V

Types of Turing machine: Turing machines and halting

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem that is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

TEXTBOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3rd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computation Science Automata languages and computation, Mishra and Chandra Sekaran, 2nd edition, PHI.

REFERENCEBOOKS:

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I. A. Cohen, John Wiley.
3. A Textbook on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
4. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.
5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.

B. Tech.

L T P C

II Year II Sem

3 0 0 3

WEB PROGRAMMING**Course Objectives:**

- Learn web design using HTML, CSS, and Java Script.
- Understand PHP for server-side scripting and databases. 3. Explore XML structure and parsing techniques.
- Study servlets, applets, and user interface components. 5. Use Word Press and Joomla for CMS development.

Course Outcomes:

- Build responsive web pages using HTML and CSS.
- Write PHP scripts for form handling and storage.
- Validate and parse XML with DTD and schemas.
- Implement servlets and applets with session management.
- Develop websites using CMS tools like Word Press.

UNIT-I

HTML- Basic Tags-List, Tables, Images, Forms, Frames, **Cascading Style Sheets (CSS)**– syntax, selectors, properties

JAVA Script- Webpage Designing using HTML, Scripting basics- Client side and server-side scripting. Java Script Object, names, literals, operators and expressions- statements and features- events - windows- documents- frames- data types- built-in functions-Browser object model-Verifying forms- HTML5- CSS3- HTML 5 canvas - Web site creation using tools.

UNIT-II

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads. Connecting to database (My SQL as reference), executing simple queries, handling results, Handling sessions and cookies File Handling in **PHP:** File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

UNIT-III

XML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemes, Document Object Model, XHTML Parsing XML Data– DOM and SAX Parsers in java. **JDBC:** JDBC Overview–JDBC implementation–Connection class–Statements–Catching Database.

UNIT-IV

APPLETS: Java applets- Life cycle of an applet Adding images to an applet Adding sound to an applet. Passing parameters to an applet. Event Handling. Introducing AWT: Working with Windows Graphics and Text. Using AWT Controls, Layout Managers and Menus. Servlet– life cycle of a servlet. The Servlet API, Handling HTTP Request and Response, using Cookies, Session Tracking. Introduction to JSP.

UNIT-V**Database Handling and Content Management System**

PHP Database Interface - Web Hosting - Content Management System - Case Study of CMS-using Word Press, Joomla- The content structure– Templates- Menu links–

Components- Modules- Text editors Permissions

TEXTBOOKS:

1. Harvey Deitel, Abbey Deitel, Internet and World Wide Web: How To Program 5th Edition.

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

INNOVATION AND ENTREPRENEURSHIP

(Common for IT, CSE(AIML), CSE(DS))

Course Objectives:

- To familiarize on the basic concepts of innovation, entrepreneurship and its importance.
- To Identify and analyze the process of problem-opportunity identification, market segmentation, and idea generation techniques.
- To initiate prototype development and understand minimum viable product.
- To develop initial Business and financial planning and Go-to-Market strategies
- To impart knowledge on establishing startups ,venture pitching and IPR

Course Outcomes:

- Understand the entrepreneurship and the entrepreneurial process and its significance in economic development.
- Assess the problem from an industry perspective and generate solutions using the design thinking principles.
- Assess market competition, estimate market size, and develop a prototype.
- Analyze Business and financial planning models and Go-to-Market strategies.
- Able to build a start-up, register IP and identify funding opportunities.

Unit I: Fundamentals of Innovation and Entrepreneurship

Innovation: Introduction, need for innovation, Features, Types of innovations, innovations in manufacturing and service sectors, fostering a culture of innovation, planning for innovation. Entrepreneurship: Introduction, types of entrepreneurship attributes, mindset of entrepreneurial and intrapreneurial leadership, Role of entrepreneurs in economic development. Woman Entrepreneurship, Importance of on-campus startups. Understanding to build entrepreneurial mindset, attributes and networks individuals while on campus. Core Teaching Tool: Simulation, Game, Industry Case Studies (Personalized for students 16 industries to choose from), VentureActivity.

Unit II: Problem and Customer Identification

Identification of gap, problem, analyzing the problem from a industry perspective, real-world problems, market and customer segmentation, validation of customer problem fit, Iterating problem-customer fit, Competition and Industry trends mapping and assessing initial opportunity, Porter's Five Force Model. Idea generation, Ideation techniques: Brainstorming, Brain writing, Round robin, and SCAMPER, Design thinking principles, Mapping of solution to problem.

Core Teaching Tool: Several types of activities including: Class, game, GenAI, 'Get out of the Building' and Venture Activity.

Unit III: Opportunity assessment and Prototype development

Identify and map global competitors, review industry trends, and understand market sizing: TAM, SAM, and SOM. Assessing scope and potential scale for the opportunity. Understanding prototyping and Minimum Viable Product(MVP).

Developing a prototype: Testing, and validation.

Core Teaching Tool: Venture Activity, no-code Innovation tools, Class activity

Unit IV: Business & Financial Models

Introduction to— Business Model and types, Lean Canvas Approach: 9-block lean canvas model, building lean canvas for your startup. Business planning: components of Business plan- Sales plan, People plan and financial plan, Financial Planning: Types of costs, preparing a financial plan for profitability using a financial template, understanding the basics of Unit economics, Economies of Scale and analyzing financial performance. Go-To-Market (GTM) approach Selecting the Right Channel, Creating digital presence, and building customer acquisition strategy.

Core Teaching Tool: Founder Case Studies– Sama and Securely Share; Class activity and discussions; Venture Activities.

Unit V: Startups and IPR

Startup requirements, building founding team members and mentors, pitch preparation, start-up registration process, funding opportunities and schemes, institutional support to entrepreneurs, startup lifecycle, documentation, legal aspects in startup, venture pitching readiness, National Innovation Startup Policy (NISP) and its features.

Patents, Designs, Patentability, Procedure for grants of patents. Indian Scenario of Patenting, International Scenario: International cooperation on Intellectual Property. Patent Rights:

Scope of Patent Rights. Copyright, trademark, and GI. Licensing and transfer of technology.

Core Teaching Tool: Expert talks; Cases; Class activity and discussions; Venture Activities.

Suggested Readings:

1. John R Bessant, Joe Tidd, Innovation and Entrepreneurship, 4E, Wiley, Latest Edition.
2. Ajay Batra, The Startup Launch Book-A Practical Guide for Launching Customer Centric Ventures, Wiley, 2020. (For Core Teaching Tool).
3. Entrepreneurship Development and Small Business Enterprises, Poornima M Charantimath, 3E, Pearson, 2018.
4. D. F. Kuratko and T. V. Rao, Entrepreneurship: A South-Asian Perspective, Cengage Learning, 2013.
5. Robert D. Hisrich, Michael P. Peters, Dean A. Shepherd, Sabyasachi Sinha (2020). Entrepreneurship, Mc Graw Hill, 11th Edition.
6. NISP-[Brochure inside pages- startup policy 2019.pdf](#)

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

DATABASE MANAGEMENT SYSTEMS LAB
(Common for CSE, IT)

Course Objectives:

- Introduce ER data model, database design and normalization
- Learn SQL basics for data definition and data manipulation

Course Outcomes:

- Design database schema for a given application and apply normalization
- Acquire skills in using SQL commands for data definition and data manipulation.
- Develop solutions for database applications using procedures, cursors and triggers

List of Experiments:

1. Concept design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. A) Querying (using ANY, ALL, UNION, INTERSECT, JOIN, Constraints etc.) B) Nested, Correlated sub queries
7. Queries using Aggregate functions, GROUPBY, HAVING and Creation and dropping of Views.
8. Triggers(Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

TEXT BOOKS:

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke,Tata McGraw Hill,3rd Edition
2. Database System Concepts, Silberschatz, Korth, McGrawHill, V edition.

REFERENCES BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S. Shahand V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M.L.Gillenson, Wiley Student Edition.

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

COMPUTER NETWORKS LAB
(Common for CSE, IT, CSE(AIML), CSE(DS))

Course Objectives:

- To understand the working principle of various communication protocols.
- To understand the network simulator environment and visualize a network topology and observe its performance
- To analyze the traffic flow and the contents of protocol frames

Course Outcomes:

- Implement data link layer framing methods
- Analyze error detection and error correction codes.
- Implement and analyze routing and congestion issues in network design.
- Implement Encoding and Decoding techniques used in presentation layer
- To be able to work with different network tools

List of Experiments

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
 2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC-CCIP
 3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
 4. Implement Dijkstra's algorithm to compute the shortest path through a network
 5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
 6. Implement distance vector routing algorithm for obtaining routing tables at each node.
 7. Implement data encryption and data decryption
 8. Write a program for congestion control using Leaky bucket algorithm.
 9. Write a program for frame sorting techniques used in buffers.
 10. **Wire shark**
 - i. Packet Capture Using Wire shark
 - ii. Starting Wire shark
 - iii. Viewing Captured Traffic
 - iv. Analysis and Statistics & Filters.
1. How to run Nmap scan
 2. Operating System Detection using Nmap
 3. Do the following using NS2 Simulator
 - I. NS2 Simulator-Introduction
 - II. Simulate to Find the Number of Packets Dropped
 - III. Simulate to Find the Number of Packets Dropped by TCP/UDP
 - IV. Simulate to Find the Number of Packets Dropped due to Congestion
 - V. Simulate to Compare Data Rate & Throughput.
 - VI. Simulate to Plot Congestion for Different Source/Destination
 - VII. Simulate to Determine the Performance with respect to Transmission of Packets

TEXTBOOK:

1. Computer Networks, Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI

REFERENCES:

1. An Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education
2. Data Communications and Networking—Behrouz A. Forouzan. 3rd Edition, TMH.

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

WEB PROGRAMMING LAB

Course Objectives:

- Learn HTML, CSS, Java Script for building responsive web interfaces.
- Understand PHP for handling forms, sessions, files, and databases.
- Explore XML structure, validation, and parsing using different techniques.
- Gain skills in database access using JDBC and SQL queries.
- Implement CMS platforms like Word Press and Joomla for dynamic websites.

Course Outcomes: Upon the completion of course, students are able to

- Design responsive websites using HTML, CSS, and Java Script effectively.
- Develop PHP applications for data input, validation, and storage.
- Manipulate and parse XML documents in web-based applications.
- Connect web applications to databases using PHP or JDBC.
- Create and manage CMS-based websites using Word Press and Joomla.

List of Experiments:

1. HTML

- A. Create a webpage with HTML describing your department. Use paragraph and list tags.
- B. Apply various colors to suitably distinguish keywords. Also apply font styling like italics, underline and two other fonts to words you find appropriate. Also use header tags.
- C. Create links on the words e.g. "Wi-Fi" and "LAN" to link them to Wikipedia pages.
- D. Insert an image and create a link such that clicking on image takes user to another page.
- E. Change the background color of the page. At the bottom create a link to take user to the top of the page.

2. HTML Tables

- A. Create a table to show your class time-table.
- B. Use tables to provide layout to your HTML page describing your university infrastructure.
- C. Use and <div> tags to provide a layout to the above page instead of a table layout.
- D. Use frames such that page is divided into 3 frames 20% on left to show contents of pages, 60% in center to show body of page, remaining on right to show remarks.
- E. Embed Audio and Video into your HTML webpage.

3. CSS

- A. Apply in-line CSS to change colors of certain text portion, bold, underline and italics certain words in your HTML webpage. Also change background color of each paragraph using in-line CSS.

- B. Write all the above styling in CSS indifferent file (.css) and link it to your webpage such that changes made in CSS file are immediately reflected on the page. Group paragraphs into single class and add styling information to the class in CSS.
 - C. Create a simple form to submit user input like his name, age, address and favourite subject, movie and singer.
 - D. Add few form elements such as radio buttons, check boxes and password field. Add a submit Button at last.
- 4. JavaScript**
- A. Create a form similar to the one in previous experiment. Put validation checks on values entered by the user using Java Script (such as age should be a value between 1 and 150).Write a Java Script program to display information box as soon as page loads.
 - B. Write a Java Script program to change back ground color after 5 seconds of page load.
 - C. Write a Java Script program to dynamically bold, italic and underline words and phrases based on user actions.
 - D. Write a JavaScript program to display a hidden div (e.g. showing stats of a player when user clicks on his name).
- 5. PHP**
- A. Design a PHP script to read user input from form controls (textboxes, radiobuttons, dropdowns) and display the submitted data.
 - B. Create a PHP program to handle file upload and store the file information in a specific directory.
 - C. Develop a PHP application to connect to a MySQL database, insert form data, and display retrieved records
 - D. Implement a PHP program to open, read, write, append, and delete text files and list all files in a directory.
- 6. XML**
- A. Create an XML document with custom tags, attributes, and define a DTD to validate its structure.
 - B. Write a PHP program to read and parse XML data using Simple XML and display it in tabular format..
 - C. Develop a PHP script to connect to a MySQL database and insert XML data into corresponding database tables
- 7. Servlets**
- A. Create user registration web Application.
 - B. At the server end, write code to retrieve contents of request object and show them to the user. Match user input password with predefined password and show “Valid User” or “Invalid User”
 - C. A web Application name as input and on submits it should show hello <name> it show start time at the right top corner of the page and provide a logout button. On clicking logout button should go to logout page shows Thank you <name> with duration of usage.
 - D. A web application that takes name and age from html page. If age less than

18 it should show Hello<name> You are not authorized to visit the site. Otherwise welcome<name>to website.

- E. A web Application name as input and on submit it should show hello <name> it show start time at the right top corner of the page and provide a logout button. On clicking logout button should go to logout page shows Thank you <name> with duration of usage.
- F. A web application that takes name and age from html page. If age less than 18 it should show Hello <name> You are not authorized to visit the site. Otherwise welcome <name >to website.
- G. Create XML Document that contain 10 users information
- H. Write a java program user Id as input and returns user details from XML file by using DOM Parser.
- I. Create XML Document that contain 10 users information
- J. Write a java program user Id as input and returns user details from XML file by using SAX Parser.

8. JDBC

- A. Establishing JDBC connection to any database (oracle/mysql,H2)
- B. Search the contents of Employee table/students table in JDBC
- C. Perform database CRED operations
- D. Using simple statements, prepared statements and callable statements

9. APPLET

- A. Create a web page using Java Script to display an image and play a sound with button-triggered events.
- B. Develop a PHP script to process HTTP form requests (GET/POST), display submitted data, and manage sessions.
- C. Write a Java Script program to change background color after 5 seconds of pageload.
- D. Write a Java Script program to dynamically bold, italic and underline words and phrases based on user actions.
- E. Write a JavaScript program to display a hidden div (e.g. showing stats of a player when user clicks on his name).

10. PHP

- A. Design a PHP script to read user input from form controls (textboxes,radiobuttons, dropdowns) and display the submitted data.
- B. Create a PHP program to handle file upload and store the file information in a specific directory.
- C. Develop a PHP application to connect to a MySQL database, insert form data, and display retrieved records
- D. Implement a PHP program to open, read, write, append, and delete text files and list all files in a directory.

11. XML

- A. Create an XML document with custom tags, attributes, and define a DTD to validate its structure.
- B. Write a PHP program to read and parse XML data using Simple XML and display it in tabular format..
- C. Develop a PHP script to connect to a MySQL database and insert XML data into corresponding database tables

12. Servlets

- A. Create user registration web Application.
- B. At the server end, write code to retrieve contents of request object and show them to the user. Match user input password with pre defined password and show "ValidUser" or "Invalid User"
- C. A web Application name as input and on submits it should show hello <name> it show start time at the right top corner of the page and provide a logout button. On clicking logout button should go to logout page shows Thank you <name> with duration of usage.
- D. A web application that takes name and age from html page. If age less than 18 it should show Hello<name> You are not authorized to visit the site. Otherwise welcome<name>to website.
- E. A web Application name as input and on submit it should show hello <name> it show start time at the right top corner of the page and provide a logout button. On clicking logout button should go to logout page shows Thank you <name> with duration of usage.
- F. A web application that takes name and age from html page. If age less than 18 it should show Hello<name> You are not authorized to visit the site. Otherwise welcome <name> to website.
- G. Create XML Document that contain 10 users information
- H. Write a java program user Id as input and returns user details from XML file by using DOM Parser.
- I. Create XML Document that contain 10 users information
- J. Write a java program user Id as input and returns user details from XML file by using SAX Parser.

13. JDBC

- A. Establishing JDBC connection to any database (oracle/mysql,H2)
- B. Search the contents of Employee table/students table in JDBC
- C. Perform database CRED operations
- D. Using simple statements, prepared statements and callable statements

14. APPLET

- A. Create a web page using Java Script to display an image and play a sound with button-triggered events.
- B. Develop a PHP script to process HTTP form requests (GET/POST), display submitted data, and manage sessions.
- C. Create a PHP-based login page that uses cookies and session tracking to display a welcome message to the user

15. DATABASE HANDLING

- A. Develop a PHP-based login and registration system using sessions and database validation
- B. Install and configure a Word Press CMS locally or on a webhost, and create pages
- B. using templates, menus, and plug in.
- C. Create a simple blog using Joomla CMS by adding components, setting permissions, and customizing modules and menus

TEXT BOOKS:

1. Harvey Deitel, Abbey Deitel, Internet and World Wide Web: How To Program 5th

Edition.

2. Java Script: The Definitive Guide by David Flanagan, O 'Reilly Media, Latest Edition.
3. Web Technologies: HTML, JavaScript, PHP, Java, JSP, ASP. NET, XML and Ajax by Uttam K. Roy, Oxford University Press
4. Michael Morrison XML Unleashed Techmedia SAMS.
5. Web Technologies, Uttam KRoy, Oxford University Press

REFERENCE BOOKS:

1. John Pollock, Javascript–ABeginners Guide, 3rd Edition--Tata McGraw-Hill Edition.
2. Keyur Shah, Gateway to Java Programmer Sun Certification, Tata Mc Graw Hill, 2002.
3. Developing Java Web Services, R. Nagappan, R. Skoczylas, R. P. Sriganesh, Wiley India.
4. Developing Enterprise Web Services ,S. Chatterjee, J. Webber, Pearson Education.
5. XML, Web Services, and the Data Revolution, F. P. Coyle, Pearson Education.
6. Building web Services with Java, 2nd Edition, S. Graham and others, Pearson Education.
7. Java Web Services, D. A. Chappell & T. Jewell, O "Reilly, SPD.
8. Mc Govern, et al., "Java web Services Architecture", Morgan Kaufmann Publishers, 2005.

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

NODEJS/REACTJS/DJANGO, UIDESIGN-FLUTTER

Prerequisites: Object Oriented Programming through Java, HTML Basics.

Course Objectives:

- To implement responsive web pages using HTML, CSS3, Bootstrap, and Java Script validation.
- To develop full-stack applications with Node.js, Express, React.js, and secure the musing JWT.
- To build and integrate RESTful APIs using Node.js and Django with frontend CRUD operations.
- To create Flutter apps with responsive layouts, state management, and custom widgets.
- To enhance Flutter apps by integrating REST APIs and adding animations for better user experience.

Course Outcomes:

- Design responsive web pages using HTML, CSS, Bootstrap, and Java Script.
- Develop backend applications using Java and Node.js.
- Build single-page applications with React.js.
- Create Flutter apps with custom widgets, layouts, and forms.
- Integrate APIs, add animations, and perform UI testing in Flutter apps.

List of Experiments: Students need to implement the following experiments

1. Build a responsive website with registration, login, catalog, and cart pages using HTML, CSS3, Bootstrap, and JavaScript validation.
2. Create a react application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages
3. Create a service in react that fetches the weather information from open weather map. Org and the display the current and historical weather information using graphical representation using chart.js
4. Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)
5. Implement JWT authentication in Node.js to create secure end points.
6. Create a TODO application in react with necessary components and deploy it into github.
7. Integrate a React frontend with Node.js backend to perform CRUD operations on a shared dataset (e.g., MongoDB, MySQL.)
8. Build a Django REST API for managing student data, tested using Postman.
9. Install Flutter and Dart SDK; Write Dart programs on data types, control flow, Functions, Class & Objects and collections to understand syntax and features.
10. Create a Flutter app show casing common widgets (Text, Image, Container, Card, List View).
11. Design a responsive UI using Row, Column, Stack, media queries, and break points.
12. Implement screen navigation using Navigator and named routes.
13. Use stateful & stateless widgets; manage state using Provider or set State
14. Design a form in Flutter within put fields, validation, and error handling (e.g., student

registration form).

15. Fetch and display data from a REST API (e.g., weather or student info) in Flutter.
16. Add basic animations (fade or slide) to a Flutter app for enhancing UI interactions

TEXT BOOK:

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.

REFERENCE BOOKS:

1. Jon Duckett, Beginning HTML, XHTML, CSS, and Java Script, Wrox Publications, 2010
2. Bryan Basham, Kathy Sierra and Bert Bates, Head First Servlets and JSP, O'Reilly Media, 2nd Edition, 2008.
3. Vasam Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, A Press.

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

INDIAN KNOWLEDGE SYSTEM

(Common for IT, CSE(AIML), CSE(DS))

Bharat is considered one of the oldest civilizations of the world. Some of the archaeological evidences proved the existence of Indus Valley Civilization in 7000 B.C. Bhartiya traditions, culture, cultural activities, rituals, sacraments, painting, art of dancing, art of singing etc. is being practised till the modern times without knowing scientific approaches behind that. Eternity of Indian knowledge system proved itself that not only many rituals but also many traditions, many streams of knowledge like astrology, mathematics, physics, chemistry, biology, language studies, yoga and meditation had been following from the starting till now with some changes, in the form of traditions.

This course is for undergraduate students to inculcate Indian values. It will promote advance study and inter disciplinary research on all aspects of the Indian knowledge system.

Course Objectives: This course aims:

- To provide a tribute of the rich culture and traditions of Indian knowledge system to students of various disciplines.
- To introduce historical account on the education and scientific literature available in ancient Indian traditions and its connections with ancient Indian Philosophy
- To give insights about the applications of Bharatiya Jnana Parampara 4. To introduce Indian approach towards health and wellbeing
- To elaborate vast contribution of ancient Indian researchers, engineers, scientists and architects to the modern world

Course Outcomes: Students will be able to:

- Understand nature, scope and related fields of Indian knowledge system.
- Demonstrate the scientific literature available in ancient Indian traditions
- Understanding the application of Bharatiya Jnana Parampara
- Understand Indian approach towards Well being
- Appreciate vast contribution of ancient Indian researchers, engineers, scientists and architects to the modern world

Unit 1: Introduction to Indian Knowledge Systems

Meaning, Nature, Scope and Salient Aspects of Bharatiya Jnana Parampara - Introduction to Vedas, Upanishads, Vidya, Kala, Jnana, Shastra - Practices and Continuity of Tradition

Unit 2: Overview of History of Indian Education and Scientific Literature

Gurukul System - Role of Sanskrit in Natural Language Processing - Scientific Literature - Vedic Literature - Available Scientific Treatises– Inter linkings

Unit 3: Introduction to Scientific Theories from Pure Sciences from Ancient Indian Knowledge Systems

Overview of theories from available ancient Indian Literature about Physics, Chemistry and Mathematics– Inter linkings and applications

Unit 4: Introduction to Ancient Indian Wellness Systems

Concept of Wellness–Yoga System- Ayurveda System-Ancient Indian Aesthetics

Unit 5: Development of Engineering, Science, Technology & Fine Arts in India

Various Industries - Silk, Cotton and Ship Building - Evolution of Indian Fine Arts Cave

and Temple Architecture, Vastu - Vidya, Sculpture, Forts and Stepwells, Observatories and Paintings - Music and Natyakala - Cultural Traditions & Folk Arts

• **Pedagogy for Teachers: Apart from Class Room Instruction, the following Methods are Suggested.**

1. Project based activities and learning.
2. Presentation and case studies.
3. Film screening and book reviews.
4. Visit to historical places, archives centre, research centre or library nearby.

Note: Activities mentioned above are only suggestive. Teacher-educators should encourage students to be innovative.

Suggested Readings:

1. B. Mahadevan, Bhat Vinayak and Nagendra Pavan R.N., (2022) 'Introduction to Indian Knowledge Systems: Concepts and Applications' PHI learning PVT, New Delhi ISBN [9789391818203]
2. Dharmapal (1971) 'Indian Science and Technology in the Eighteenth Century'. Other India Press, Goa.
3. Kapil Kapoor, Singh Avdhesh Kumar, (2005) 'Indian Knowledge Systems' D.K. Printworld (P) Ltd. ISBN 10: 8124603367 / ISBN 13: 9788124603369
4. Chakradeo Ujwala, Temples of Bharat, Aayu Publications, New Delhi, 2024.
5. D.N.Bose, S.N. Senand B.V. Subbarayappa, A Concise History of Science in India, Indian National Science Academy, New Delhi, 2009.
6. Datta B. and A.N. Singh, *History of Hindu Mathematics: Parts I and II*, Asia Publishing House, Bombay, 1962.
7. Kapoor, K. (2021), *Indian Knowledge System: Nature, Philosophy, Character in Indian Knowledge System*, vol. 1, Pub. Indian Institute of Advanced Studies, Shimla
8. Mahadevan, B., Bhat, V.R., Pavana, N. (2022), Philosophical Systems, in Introduction to Indian Knowledge System, Pub. PHI Learning, New Delhi.
9. Mahadevan, B., Bhat, V.R., Pavana, N. (2022), Knowledge: Framework and Classification, in Introduction to Indian Knowledge System, Pub. PHI Learning, New Delhi.

Video Resources:

1. Introductory lectures by Prof. Gauri Mahulikar
2. Introductory lectures by Prof. Kapil Kapoor

Websites:

- <https://iksindia.org/index.php>
- Official Website of IKS-Indian Knowledge System
- <https://www.youtube.com/watch?v=uKcf-hSlcUE>
- Address by Prof Kapil Kapoor | Indian Institute of Advanced Study (FDP2021)
- https://www.youtube.com/watch?v=MDJTXNiH2_A
- Mukul Kanitkar on Bharatiya Knowledge System
- <https://www.youtube.com/watch?v=uARMhv97pjk>
- <https://www.youtube.com/watch?v=oTwwgf56GbsA>
- Scientific History of India | Mukul Kanitkar Lecture in DTU
- <https://youtu.be/gNjNmPJqXJc?si=WFBbuUT65mLZzpOW>
- Ancient India's Scientific Achievements & Contribution in Mathematics, Astronomy, Science & Medicine.