



SPHOORTHY

ENGINEERING COLLEGE

(UGC AUTONOMOUS)

B.Tech. in COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)
Course Structure & Syllabus (SR25 Regulations)
Applicable from AY 2025-2026 Batch

II YEAR I SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1.	S25MA301BS	Mathematical and Statistical Foundations	3	0	0	3
2.	S25CY302PC	Computer Organization	3	0	0	3
3.	S25CY303PC	Java Programming	3	0	0	3
4.	S25CS304PC	Software Engineering	3	0	0	3
5.	S25CS305PC	Database Management System	3	0	0	3
6.	S25MA306PC	Computational Mathematics Lab	0	0	2	1
7.	S25CY307PC	Java Programming Lab	0	0	2	1
8.	S25CS308PC	Software Engineering Lab	0	0	2	1
9.	S25CS309PC	Database Management Systems Lab	0	0	2	1
10.	S25CY310SD	Data Visualization-R Programming/Power BI/ Tableau/Google Chart	0	0	2	1
Total Credits			15	0	10	20

S.No.	Course Code	Course Title	L	T	P	Credits
1.	S25CS401PC	Discrete Mathematics	3	0	0	3
2.	S25CS402PC	Operating Systems	3	0	0	3
3.	S25CY403PC	Formal Languages and Automata Theory	3	0	0	3
4.	S25CS404PC	Computer Networks	3	0	0	3
5.	S25CY405PC	Mathematical Foundations of Cryptography	3	0	0	3
6.	S25MS406HS	Innovation and Entrepreneurship	2	0	0	2
7.	S25CS407PC	Operating Systems Lab	0	0	2	1
8.	S25CS408PC	Computer Networks lab	0	0	2	1
9.	S25CY409PC	Mathematical Foundations of Cryptography Lab	0	0	2	1
10.	S25CY410SD	Node JS/ React JS/ Django, UI Design - Flutter	0	0	2	1
11.	S25VA400HS	Indian Knowledge System	1	0	0	1
Total Credits			18	0	08	22

S25MA301BS: MATHEMATICAL AND STATISTICAL FOUNDATIONS**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

Pre-requisites: Mathematics courses of first year of study.**Objectives:** To learn

1. The Number Theory basic concepts useful for cryptography etc.
2. The theory of Probability, and probability distributions of single random variables.
3. The sampling theory and testing of hypothesis and making inferences.
4. 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

1. Apply the number theory concepts to cryptography domain.
2. Apply the concepts of probability and distributions to some case studies.
3. Correlate the material of one unit to the material in other units.
4. 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**10 L**

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**8 L**

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**10 L**

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 (Large and Small Samples)**10 L**

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**10 L**

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 & Statistics for Engineers & Scientists, 9th Ed. Pearson Publishers.
3. S C Gupta and V K Kapoor, Fundamentals of Mathematical Statistics, Khanna publications.

REFERENCE BOOKS:

1. T.T. Soong, Fundamentals of Probability and Statistics for Engineers, John Wiley & Sons, Ltd, 2004.
2. Sheldon M Ross, Probability and statistics for Engineers and scientists, academic press.
3. S C Gupta and V K Kapoor, Fundamentals of Mathematical statistics, Khanna publications.

S25CY302PC: COMPUTER ORGANIZATION**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

Course Outcomes:

1. Demonstrate the ability to represent and manipulate data using binary number systems and apply Boolean algebra to analyze and design digital logic circuits.
2. Simplify Boolean functions using Karnaugh Maps and design minimal logic circuits using universal logic gates.
3. Classify data representation formats and implement arithmetic, logic, and shift microoperations using register transfer language.
4. Illustrate the basic organization of a computer and differentiate between hardwired and microprogrammed control unit designs.
5. Analyze CPU structure, addressing modes, and execute arithmetic operations including fixed- point, floating-point, and BCD formats.

UNIT - I

Binary Systems: Digital Systems, Binary Numbers, Number Base Conversions, Octal and Hexadecimal Numbers, Complements, Signed Binary Numbers, Binary Codes, Binary Storage and Registers, Binary Logic.

Boolean Algebra And Logic Gates: Basic Definitions, Axiomatic definition of Boolean Algebra, Basic theorems and properties of Boolean Algebra, Boolean Functions, Canonical and Standard Forms, Other Logic Operations, Digital Logic Gates, Integrated Circuits.

UNIT - II

Gate-Level Minimization: The Map method, Four-variable map, Five-variable map, Product of Sum's simplifications, Don't care conditions, NAND and NOR implementation, other two level implementations, Exclusive-OR Function.

UNIT - III

Basic Structure of Computers: Computer Types, Functional unit, Data Representation, Fixed Point Representation, Floating Point Representation, Error Detection codes.

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, Logic micro-operations, Shift micro- operations, Arithmetic logic shift unit.

UNIT - IV

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

Micro Programmed Control: Control memory, Address sequencing, micro program example, design of control unit, Micro program Sequencer, Hard wired control Vs Micro programmed control

UNIT - V

Central Processing Unit Organization: General Register Organization, STACK organization. Instruction formats, Addressing modes. DATA Transfer and manipulation, Program control. Reduced Instruction set computer.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Floating - point Arithmetic, BCD Adder.

TEXT BOOKS:

1. Digital Design: With an Introduction to the Verilog HDL - Fifth Edition, M. Morris Mano, Pearson

Education.

2. Fundamentals of Logic Design - Roth, 7th Edition, Thomson.
3. Computer Systems Architecture - M. Moris Mano, 3rd Edition, Pearson/PHI
4. Computer Organization - Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5th Edition, McGraw Hill.

REFERENCE BOOKS:

1. Digital Principles and Design - Donald D. Givone, Tata Mc Graw Hill.
2. Fundamentals of Digital Logic and Micro Computer Design, 5th Edition, M. Rafiquzzaman, Wiley-Interscience.
3. Computer Organization and Architecture - William Stallings 7th Edition, Pearson/ PHI.
4. Structured Computer Organization - Andrew S. Tanenbaum, 6th Edition PHI/Pearson.

S25CY303PC: JAVA PROGRAMMING**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

Course Outcomes:

1. Identify the model of Object-Oriented Programming: Abstract data types, Encapsulation, Inheritance and Polymorphism.
2. Summarize the fundamental features like Interfaces, Exceptions and Collections.
3. Correlate the advantages of Multi-threading.
4. 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**MULTITHREADING, 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: ArrayList, Vector, TreeSet, HashMap, Hashtable, 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, Split Pane, 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.

S25CS304PC: SOFTWARE ENGINEERING**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

Course Objectives

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

Course Outcomes

- Ability to translate end-user requirements into system and software requirements, using e.g.
- UML, and structure the requirements in a Software Requirements Document (SRD).
- Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

UNIT - I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI). Process models: The waterfall model, Spiral model, Incremental Process Models, Concurrent Models, Component based development and Agile Development.

UNIT - II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.
Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT - III

Design Engineering: Design process and design quality, design concepts, the design model.
Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, use case diagrams, class diagrams, sequence diagrams, collaboration diagrams, activity diagrams and component diagrams.

UNIT - IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.
Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT - V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.

2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS:

1. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.
4. Fundamentals of Software Engineering-Rajib Mall, PHI.

S25CS305PC: DATABASE MANAGEMENT SYSTEMS**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

Prerequisites: A course on “Data Structures”.**Course Objectives:**

1. To understand the basic concepts and the applications of database systems.
2. To master the basics of SQL and construct queries using SQL.
3. Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

Course Outcomes:

1. Gain knowledge of fundamentals of DBMS, database design and normal forms
2. Master the basics of SQL for retrieval and management of data.
3. 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.

TEXT BOOKS:

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

REFERENCE 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.Shah and 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.

S25MA306PC: COMPUTATIONAL MATHEMATICS LAB
(Using Python/MATLAB software)

B.Tech. II Year I Sem.

L T P C
0 0 2 1

Pre-requisites: Matrices, Iterative methods and ordinary differential equations

Course Objectives: To learn

1. Solve problems of Eigen values and Eigen Vectors using Python/MATLAB.
2. Solution of Algebraic and Transcendental Equations using Python/MATLAB
3. Solve problems of Linear system of equations
4. 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

1. Develop the code to find the Eigen values and Eigen Vectors using Python/MATLAB.
2. Develop the code find solution of Algebraic and Transcendental Equations and Linear system of equations using Python/MATLAB
3. 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 Eigenvectors:

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-Seidal iteration 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 coefficients

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, Martin Novak 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.

S25CY307PC: JAVA PROGRAMMING LAB**B.Tech. II Year I Sem.**

L	T	P	C
0	0	2	1

Course Outcomes:

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

TASK 1

Write java programs that implement the following

- a. Class and object
- b. Constructor
- c. Parameterized constructor
- d. Method overloading
- e. Constructor overloading.

TASK 2

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

TASK 3

Write java programs that uses the following keywords

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

TASK 4

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

TASK 5

- a) 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.
- b) Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
- c) Write a Java program that displays the number of characters, lines and words in a text file

TASK 6

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

TASK 7

- a) 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 interthread communication.

TASK 8

Write a program illustrating following collections framework

- a) ArrayList b) Vector c) Hash Table d) Stack

TASK 9

- 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.

TASK 10

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

TASK 11

- 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 Num1 or 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.

TASK 12

- 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, 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 (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.

S25CS308PC: SOFTWARE ENGINEERING LAB**B.Tech. II Year I Sem.**

L	T	P	C
0	0	2	1

Prerequisites: A course on “Programming for Problem Solving”.**Co-requisite:** A Course on “Software Engineering”.**Course Objectives:**

- To have hands-on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

Course Outcomes:

- Ability to translate end-user requirements into system and software requirements
- Ability to generate a high-level design of the system from the software requirements
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

List of Experiments

Do the following seven exercises for any two projects given in the list of sample projects or any other Projects:

1. Development of problem statements.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Preparation of Software Configuration Management and Risk Management related documents.
4. Study and usage of any Design phase CASE tool
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various white box and black box testing techniques.

Sample Projects:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS:

1. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill

S25CS309PC: DATABASE MANAGEMENT SYSTEMS LAB**B.Tech. II Year I Sem.**

L	T	P	C
0	0	2	1

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 subqueries
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. General Functions(Date, Character,Number)
9. Triggers (Creation of insert trigger, delete trigger, update trigger)
10. Procedures
11. Usage of Cursors

TEXT BOOKS:

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill,3rd Edition
2. Database System Concepts, Silberschatz, Korth, McGraw Hill, 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. Shah and 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.

S25CY310SD: DATA VISUALIZATION - R PROGRAMMING/ POWER BI/TABLEAU/GOOGLE CHART**B.Tech. II Year I Sem.**

L	T	P	C
0	0	2	1

Course Objectives:

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

Course Outcomes:

1. Understand how to import data into Tableau.
2. Understand concepts of Dimensions and Measures.
3. Develop Programs and understand how to map Visual Layouts and Graphical Properties
4. 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 from Kaggle.
12. Visualize data patterns using Google Charts by creating interactive Line, Bar, and Pie charts along with Dashboards using HTML and JavaScript.

REFERENCE BOOKS:

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

WEB RESOURCE:

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

II YEAR II SEMSTER

S.No.	Course Code	Course Title	L	T	P	Credits
1.	S25CS401PC	Discrete Mathematics	3	0	0	3
2.	S25CS402PC	Operating Systems	3	0	0	3
3.	S25CY403PC	Formal Languages and Automata Theory	3	0	0	3
4.	S25CS404PC	Computer Networks	3	0	0	3
5.	S25CY405PC	Mathematical Foundations of Cryptography	3	0	0	3
6.	S25MS406HS	Innovation and Entrepreneurship	2	0	0	2
7.	S25CS407PC	Operating Systems Lab	0	0	2	1
8.	S25CS408PC	Computer Networks lab	0	0	2	1
9.	S25CY409PC	Mathematical Foundations of Cryptography Lab	0	0	2	1
10.	S25CY410SD	Node JS/ React JS/ Django, UI Design - Flutter	0	0	2	1
11.	S25VA400HS	Indian Knowledge System	1	0	0	1
		Total Credits	18	0	08	22

S25CS401PC: DISCRETE MATHEMATICS**B.Tech. II Year II Sem.**

L	T	P	C
3	0	0	3

Course Objectives:

1. Introduces elementary discrete mathematics for computer science and engineering.
2. 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:

1. Understand and construct precise mathematical proofs
2. Apply logic and set theory to formulate precise statements
3. Analyze and solve counting problems on finite and discrete structures
4. Describe and manipulate sequences
5. 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 Subgraphs, 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.

TEXT BOOKS:

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: Joe I. Mott, Abraham Kandel, Theodore P. Baker, Prentis Hall of India, 2nd ed.

REFERENCE BOOKS:

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

S25CS402PC: OPERATING SYSTEMS**B.Tech. II Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisites:

1. A course on "Computer Programming and Data Structures".
2. A course on "Computer Organization and Architecture".

Course Objectives:

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

Course Outcomes:

1. Will be able to control access to a computer and the files that may be shared
2. Demonstrate the knowledge of the components of computers and their respective roles in computing.
3. Ability to recognize and resolve user problems with standard operating environments.
4. 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, Multiprogrammed, 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 **Interprocess 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 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.

S25CY403PC: FORMAL LANGUAGES AND AUTOMATA THEORY**B.Tech. II Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisites: Mathematical Foundations**Course Objectives**

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

Course Outcomes

1. Understand the concept of abstract machines and their power to recognize the languages.
2. Employ finite state machines for modeling and solving computing problems.
3. 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 Tree, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

Push Down Automata: Definition of the Pushdown Automaton, 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.

TEXT BOOKS:

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

REFERENCE BOOKS:

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 Text book 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.

S25CS404PC: COMPUTER NETWORKS**B.Tech. II Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisites

1. A course on "Programming for problem solving".
2. A course on "Data Structures".

Course Objectives

1. Equip the students with a general overview of the concepts and fundamentals of computer networks.
2. Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.
3. Elucidate the students about working and implementation of protocols at various layers in protocols stack.
4. Appreciating the protocol working by observing and analysing outputs of the packet sniffer,

Course Outcomes

1. Gain the knowledge of the basic computer network technology.
2. Gain the knowledge of the functions of each layer in the ISO-OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.
5. Understanding working of the protocols through traces captured by a packet sniffer

UNIT - I

Introduction: The Internet, Protocol, Network Edge, Access Networks, Network Core, Packet Switching, Circuit Switching, Delay, Loss, and Throughput in Packet-Switched Networks, Protocol reference models: ISO-OSI, TCP/IP, Types of Network attacks, History of Computer Networking and the Internet.

UNIT-II

The Physical and Data Link Layer:

Physical Layer: Guided Transmission Media: Twisted pairs, Coaxial cable, fiber optics, Wireless Transmission.

Datalink Layer: The Services Provided by the Link Layer, Error-Detection and -Correction Techniques- Parity Checks, Checksum Methods, Cyclic Redundancy Check (CRC), Hamming code, Multiple Access Links and Protocols, Channel Partitioning Protocols, Random Access Protocols, Taking-Turns Protocols, DOCSIS: The Link-Layer Protocol for Cable Internet Access, Switched Local Area Networks, Link-Layer Addressing and ARP, Ethernet, Link-Layer Switches, Virtual Local Area Networks (VLANs), Link Virtualization-Multiprotocol Label Switching (MPLS), Data Center Networking, A Day in the Life of a Web Page Request. Wireless network characteristics, Wireless LAN.

UNIT - III

Network Layer: Data and Control plane, Forwarding and Routing 308, Network Service Models, Virtual Circuit and Datagram Networks, Router working, The Internet Protocol (IP): Forwarding and Addressing in the Internet, Datagram Format, IPv4 Addressing, Internet Control Message Protocol (ICMP), IPv6, IP Security, Routing Algorithms- The Link-State (LS) Routing Algorithm, The Distance-Vector (DV) Routing Algorithm, Hierarchical Routing, Routing in the Internet-Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet: OSPF, Inter-AS Routing: BGP, Broadcast and Multicast Routing, Broadcast Routing Algorithms, Multicasting.

UNIT - IV

Transport Layer: Transport-Layer Services, Multiplexing and Demultiplexing, Connectionless Transport: UDP, Principles of Reliable Data Transfer, Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocols, Go-Back-N (GBN), Selective Repeat (SR), Connection-Oriented Transport: TCP, The TCP Connection, Segment Structure, Round-Trip Time Estimation and Timeout, Reliable Data Transfer, Flow Control, TCP Connection Management,

Principles of Congestion Control, TCP Congestion Control, Fairness.

UNIT - V

Application Layer: Principles of Network Applications, Network Application Architectures, Processes Communicating, Transport Services Available to Applications, Transport Services Provided by the Internet, Application-Layer Protocols, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, SMTP, DNS, Peer-to-Peer Applications, Socket Programming: Creating Network Applications.

TEXT BOOKS:

1. Computer Networking: A Top-Down Approach - James F.Kurose, Keith W. Ross, Pearson
2. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson/PHI

REFERENCE BOOK:

1. Data Communications and Networking - Behrouz A. Forouzan. Third Edition TMH.

S25CY405PC: MATHEMATICAL FOUNDATIONS OF CRYPTOGRAPHY**B.Tech. II Year II Sem.**

L	T	P	C
3	0	0	3

Course Objectives

1. Build a solid mathematical basis to understand foundations of cryptography
2. Formally understand the notions related to security authentication and privacy.
3. Provide a rigorous treatment of the emerging and key subject subarea of CSE - security.

Course Outcomes

1. Students will gain an understanding of cryptosystems widely used to protect data security on the internet, and be able to apply the ideas in new situations as needed.

UNIT - I

Basic functions of cryptography - Encryption Schemes, Digital Signatures, Fault Tolerant Protocols and Zero-Knowledge Proofs.

The Computational Model: P, NP, and NP- Completeness, Probabilistic Polynomial Time, Non-Uniform Polynomial Time

UNIT - II**Computational Difficulty**

One-Way Functions Definitions, Strong One- Way Functions, Weak One-Way Functions, Universal One-Way Function, Trapdoor One-Way Permutations Computational Indistinguishability: Definition, Relation to Statistical Closeness, Indistinguishability by Repeated Experiments, Indistinguishability by Circuits

UNIT - III**Zero-Knowledge Proof Systems**

Zero-Knowledge Proofs, Perfect and Computational Zero-Knowledge, An Example (Graph Isomorphism in PZK) Zero-Knowledge with Respect to Auxiliary Inputs

UNIT - IV**Encryption Schemes**

Private-Key versus Public-Key Schemes, The Syntax of Encryption Schemes, Semantic Security, Indistinguishability of Encryptions, Stream--Ciphers, Preliminaries: Block--Ciphers

UNIT - V

Digital Signatures and Message Authentication: Attacks and security, Variants

Constructions of Message Authentication Schemes: Applying a pseudorandom function to the document

TEXTBOOK:

1. Foundations of Cryptography (two volumes), Oded Goldreich, Cambridge university Press, 2004. (Indian print available).

REFERENCES:

1. Introduction to Modern Cryptography, J. Katz, Y. Lindell, Chapman Hall, USA 2007.
2. Modern cryptography - Theory and practice, Wen Bo Mao, Prentice Hall, USA, 2003 (Indian edition available)

S25MS406HS: INNOVATION AND ENTREPRENEURSHIP**B.Tech. II Year II Sem.**

L	T	P	C
2	0	0	2

Course Objectives:

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

Course Outcomes:

1. Understand the entrepreneurship and the entrepreneurial process and its significance in economic development.
2. Assess the problem from an industry perspective and generate solutions using the design thinking principles.
3. Assess market competition, estimate market size, and develop a prototype.
4. Analyze Business and financial planning models and Go-to-Market strategies.
5. 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), Venture Activity.

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, Gen AI, '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, McGrawHill, 11th Edition.
6. NISP -[Brochure inside pages - startup policy 2019.pdf](#)

S25CS407PC: OPERATING SYSTEMS LAB**B.Tech. II Year II Sem.**

L	T	P	C
0	0	2	1

Prerequisites:

- A course on “Programming for Problem Solving”.
- A course on “Computer Organization and Architecture”.

Co-requisite: A course on “Operating Systems”.**Course Objectives:**

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

Course Outcomes:

1. Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
2. 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 Deadlock 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 a) FCFS b) LRU c) Optimal
8. Simulate all File Organization Techniques a) Single level directory b) Two level directory

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.

S25CS408PC: COMPUTER NETWORKS LAB**B.Tech. II Year II Sem.**

L	T	P	C
0	0	2	1

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. **Wireshark**
 - 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

TEXT BOOK:

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.

S25CY409PC: MATHEMATICAL FOUNDATIONS OF CRYPTOGRAPHY LAB**B.Tech. II Year II Sem.**

L	T	P	C
0	0	2	1

1. Implement Basic Encryption/Decryption Schemes

- Implement Caesar cipher, Vigenère cipher, and XOR cipher.
- Test encryption and decryption on text input.

2. Demonstrate Digital Signature Generation and Verification

- Use Python libraries (e.g., cryptography or pycryptodome) to sign a message with a private key and verify it using a public key.

3. Simulate a Fault-Tolerant Protocol

- Simulate a basic consensus protocol (like 2-phase commit) handling failure scenarios.

4. Implement Zero-Knowledge Proof (ZKP) Concept

- Simulate a ZKP for Graph Isomorphism using a challenge-response protocol.

5. Implement a Simple One-Way Hash Function

- Build a toy hash function and demonstrate its one-way property. Compare it to SHA-256.

6. Test for Strong and Weak One-Way Functions

- Write a program to simulate strong vs weak one-way functions. Use time complexity analysis to show difficulty of inversion.

7. Demonstrate Trapdoor One-Way Permutations

- Use RSA key generation to demonstrate how trapdoor functions work in public-key encryption.

8. Demonstrate Computational Indistinguishability

- Create two similar distributions and test whether a computational process can distinguish between them (simulate using coin tosses).

9. Implement a Zero-Knowledge Proof for Discrete Logarithms

- Given a number $y = g^x \bmod p$, simulate ZKP to prove knowledge of x without revealing it.

10. Implement Perfect and Computational ZKP

- Demonstrate perfect ZKP using small problems like Sudoku validation without revealing the solution.

11. Auxiliary Inputs in ZKP

- Simulate a scenario where an adversary has auxiliary information and test how ZKP maintains secrecy.

12. Compare Private-Key vs Public-Key Encryption

- Implement AES (symmetric) and RSA (asymmetric) and compare their usage on messages.

13. Demonstrate Block Cipher Modes

- Implement ECB, CBC, and CFB modes for AES and show their impact on ciphertext for patterned plaintext.

14. Implement a Digital Signature Scheme Using RSA

- Generate keys, sign a message, and verify using the RSA algorithm.

15. Construct a Message Authentication Code (MAC)

- Use a pseudorandom function like HMAC-SHA256 to generate and verify MACs for file integrity.

TEXT BOOK:

1. Foundations of Cryptography (two volumes), Oded Goldreich, Cambridge university Press, 2004. (Indian print available).

REFERENCE BOOKS:

1. Introduction to Modern Cryptography, J. Katz, Y. Lindell, Chapman Hall, USA 2007.
2. Modern cryptography - Theory and practice, Wen Bo Mao, Prentice Hall, USA, 2003 (Indian edition available).

S25CY410SD: NODE JS/ REACT JS/ DJANGO, UI DESIGN - FLUTTER**B.Tech. II Year I Sem.**

L	T	P	C
0	0	2	1

Prerequisites: Object Oriented Programming through Java, HTML Basics.**Course Objectives:**

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

Course Outcomes:

1. Design responsive web pages using HTML, CSS, Bootstrap, and JavaScript.
2. Develop backend applications using Java and Node.js.
3. Build single-page applications with React.js.
4. Create Flutter apps with custom widgets, layouts, and forms.
5. 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 openweathermap.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 endpoints.
- 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 showcasing common widgets (Text, Image, Container, Card, ListView).
- 11 Design a responsive UI using Row, Column, Stack, media queries, and breakpoints.
- 12 Implement screen navigation using Navigator and named routes.
- 13 Use stateful & stateless widgets; manage state using Provider or setState.
- 14 Design a form in Flutter with input 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 JavaScript, Wrox Publications, 2010
2. Bryan Basham, Kathy Sierra and Bert Bates, Head First Servlets and JSP, O'Reilly Media, 2nd Edition, 2008.
3. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, A Press.

S25VA400HS: INDIAN KNOWLEDGE SYSTEM**B.Tech. II Year II Sem.**

L	T	P	C
1	0	0	1

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:

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

Course Outcomes: Students will be able to:

1. Understand nature, scope and related fields of Indian knowledge system.
2. Demonstrate the scientific literature available in ancient Indian traditions
3. Understanding the application of Bharatiya Jnana Parampara
4. Understand Indian approach towards Wellbeing
5. 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 - Interlinkings

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 - Interlinkings 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. Sen and 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://iksin dia.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 (FDP 2021)
- 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=oTwgf56GbsA>
- 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