Syllabus – Sem III and Sem IV

- ❖ Computer Science and Engineering (CS)
- ❖ Computer Engineering (KE)
- ❖ Artificial Intelligence & Data Science (AIDS)
- ❖ Computer Science and Engineering (Data Science) (DS)

SEM III

3CS200PC/3DS200PC/3AD200PC/3KE200PC

DISCRETE STRUCTURE AND GRAPH THEORY

Course Pre-requisite:
Basic knowledge of Mathematics

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of Discrete Structure by being able to do each of the following:

- Use mathematically correct terminology and notation.
- Constructs correct direct and indirect proofs.
- Apply logical reasoning to solve a variety of problems

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:

1. Analyze and express logic sentences in terms of predicates, quantifiers, and logical connectives.
2. Derive the solution for a given problem using deductive logic and prove the solution based on logical inference.
3. Classify algebraic structure for a given mathematical problem.
4. Perform combinatorial analysis to solve counting problems.
5. Perform operation on trees data structures.
6. Develop the given problem as graph networks and solve with techniques of graph theory

Syllabus:

Unit I: The Foundations: Logic and Proofs (Hours: 07)
Propositions, Truth Tables, Compound Propositions, Logical Operators, Logic and Bit Operations; Logical Equivalences, Normal Forms, De Morgan's Laws, Satisfiability: Applications and Solving Problems; Predicates, Quantifiers: Restricted Domains, Precedence, Logical Equivalences; Rules of Inference for Propositional Logic.

Unit II: Sets, Functions and Relation (Hours: 07)
Introduction, Venn Diagrams, Subsets, Size of a Set, Power Sets, Cartesian Products, Set Notation with Quantifiers, Truth Sets and Quantifiers, Set Operations ,Functions, Inverse

Functions, Compositions and Graphs of Functions ,Partial Functions; Sequences, Summations; Countable Sets, An Uncountable Set; Functions as Relations, Relations on a Set, Properties of Relations, Combining Relations; Representing Relations Using Matrices; Representing Relations, Closures of Relations, Equivalence Relations.

Unit III: Algebraic Structures (Hours: 07)

Algebraic Systems: Examples and General Properties; Semigroups and Monoids: Homomorphism of Semigroups and Monoids, Subsemigroups and Submonoids; Groups: Definitions, Subgroups and Homomorphisms, Cosets and Lagrange's Theorem, Normal Subgroups, algebraic Systems with Two Binary Operations; Group Codes: The Communication Model and Basic Notions of Error Correction, Hamming Distance.

Unit IV: Boolean Algebra (Hours: 07)

Lattices, Boolean Algebra: Boolean Functions, Representing Boolean Functions, sum of product expansions, Product of sum expansion Functional Completeness, Logic Gates, Combinations of Gate, Minimization of Circuits, Karnaugh Maps.

Unit V: Tree (Hours: 07)

Introduction, Rooted Tree, ordered rooted tree, tree as model, Properties of Trees, Applications of tree, Binary Search Trees, Decision Trees, Prefix Codes, Huffman Coding, Game Trees, Tree traversal, Preorder Traversing, Inorder Traversing, Post order Traversing, Spanning Tree, Minimum spanning tree

Unit VI: Graph (Hours: 07)

Graph Models; Basic Terminology, Special Simple Graphs, Bipartite Graphs, Matchings, Applications of Special Types of Graphs, New Graphs from Old; Graph Representation, Adjacency and Incidence Matrices, Isomorphism of Graphs, Determining Isomorphism; Paths, Connectedness in Undirected Graphs and Directed Graphs, Paths and Isomorphism, Counting Paths Between Vertices; Euler Paths and Circuits, Hamilton Paths and Circuits, Applications of Hamilton Circuits; Planar Graphs: Euler's Formula, Kuratowski's Theorem; Graph Coloring: Introduction, Applications of Graph Coloring

Text Books:

1. Kenneth H. Rosen: Discrete Mathematics and Its Applications, 7th Edition, McGraw-Hill.
2. J. P. Tremblay and R. Manohar: Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw-Hill Edition, McGraw-Hill.

Reference Books:

1. Norman L. Biggs: Discrete Mathematics, 2nd Edition, Oxford University Press.
2. Seymour Lipschutz and Marc Lars Lipson: Schaum's Outline of Theory and Problems of Discrete Mathematics, 3rd Edition, Schaum's Outlines Series, McGraw-Hill.
3. C. L. Liu and D. P. Mohapatra: Elements of Discrete Mathematics: A Computer Oriented Approach, 3rd Edition, Tata McGraw-Hill, McGraw-Hill.

== XX ==

3CS201PC/3AD201PC

OBJECT ORIENTED PROGRAMMING

Course Prerequisite:
Computer Programming

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding:

- To explore the principles of Object Oriented Programming (OOP) such as data abstraction, encapsulation, inheritance and polymorphism.
- To use the object-oriented paradigm in program design.
- To Provide programming insight using OOP constructs.
- To lay a foundation for advanced programming

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to

1. Apply Object Oriented approach to design software.
2. Implement programs using classes and objects.
3. Specify the forms of inheritance and use them in programs.
4. Analyze polymorphic behavior of objects.
5. Design and develop GUI programs.
6. Develop Applets for web applications

Syllabus:

Unit I: Introduction to Object Oriented Programming: (Hours:07)
Introduction, Need of OOP, Principles of Object-Oriented Languages, Procedural Language Vs OOP, Application of OOP, Java Virtual Machine, Java features, Program Structures. Java Programming Constructs: Variables, Primitive data types, Identifier, Literals, Operators, Expressions, Precedence Rules and Associativity, Primitive Type Conversion and Casting, Flow of Control.

Unit II: Classes and Objects: (Hours:07)
Classes, Objects, Creating Objects, Methods, Constructors, Cleaning up Unused Objects, Class Variable and Methods, this keyword, Arrays, Command Line Arguments.

Unit III: Inheritance, Interfaces and Packages: (Hours:07)
Inheritance: Inheritance vs. Aggregation, Method Overriding, super keyword, final keyword, Abstract class. Interfaces: Defining interfaces, Implementing interfaces, Accessing interface variables, Extending interfaces. Packages: Packages, java.lang package

Unit IV: Exception handling and Input/Output: (Hours:07)
Exception: Introduction, Exception handling Techniques, User-defind exception, Exception Encapsulation and Enrichment. Input/Output: The java.io.file Class, Reading

and Writing data, Randomly Accessing a file, Reading and Writing Files using I/O Package.

Unit V: Applets: (Hours:07)
Introduction, Applet Class, Applet structure, Applet Life cycle, Common Methods used in displaying the output, paint (), update () and repaint (), More about applet tag, getDocumentBase() and getCodeBase () methods, Applet Context Interface, Audio clip, Graphic Class, Color, Font, Font Metrics.

Unit VI: Event Handling: (Hours:07)
Introduction, Event delegation Model, java.awt.event Description, Sources of events, Event Listeners, Adapter classes, Inner Classes. Abstract Window Toolkit: Introduction, Components and Containers, Button, Label, Checkbox, Radio Buttons, List Boxes, Choice Boxes, Textfield and Textarea, Container Class, Layouts, Menu, Scrollbar.

Text Books:
1. Sachin Malhotra and Saurabh Choudhary: Programming in Java, Oxford University Press 2010.
2. Herbert Schildt: Java Complete References (McGraw Hill)

Reference Books:
1. H.M.Dietel and P.J.Dietel, "Java How to Program" Pearson Education/PHI, Sixth Edition.
2. E. Balagurusamy: Programming with Java (McGraw Hill)
3. Dr. R. NageswaraRao: Core Java An Integrated Approach (Dreamtech)
4. Khalid Mughal: A Programmer"s Guide to Java Certification, 3rdEdition (Pearson)
5. Sharnam Shah and Vaishali Shah: Core Java for Beginners,(SPD),2010.

== XX ==

3DS201PC/3KE201PC

PROGRAMMING METHODOLOGY USING PYTHON

Course Pre-requisites:
Basic Knowledge of Programming fundamentals

Course Objective:
Throughout the course, students will be expected to demonstrate their understanding of: Programming Methodology Using Python by being able to do each of the following:
- Describe the core syntax and semantics of Python programming language.
- Discover the need for working with the strings and functions.
- Illustrate the process of structuring the data using lists, dictionaries, tuples and sets.
- Indicate the use of modules, packages and built-in functions to navigate the file system.
- Infer the Object-oriented Programming concepts in Python.
- To develop the ability to write database applications in Python.

Course Outcomes:

On completion of the course, the students will be able to

1. Apply various fundamentals for problem solving using python
2. Develop proficiency in creating applications using the Python programming Language.
3. Understand the various data structures available in Python programming language and apply them in solving computational problems.
4. Draw various kinds of plots.

Syllabus:

Unit I: Introduction: (Hours:07)

Basic concepts of Python- Variables, Data Types, Operators: Arithmetic, Logical, Relational, Conditional Statements: if, if-else, elif, Looping Statements: for, while, Control Statements: break, continue, pass.

Unit II : Strings and Functions: (Hours:07)

String Manipulation-Accessing Strings- Basic Operations-String slices- Function and Methods Functions Defining a function- Calling a function- Types of functions Function Arguments-Anonymous functions- Global and local variables

Unit III: Data Structures in Python: (08 Hrs)

Lists-Introduction - Accessing list -Operations - Working with lists - Function and Method Tuple -Introduction Accessing – Tuples – Operations – Working - Functions and Methods Dictionaries -Introduction - Accessing values in dictionaries - Working with dictionaries Properties - Functions.

Unit IV : Classes and objects: (Hours:07)

Overview of OOP, Class Definition, Creating Objects, Objects as Arguments, Objects as Return Values, Built-in Class Attributes, Inheritance- Overloading- Overriding- Data hiding.

Unit V: Modules and Packages: (Hours:07)

Standard modules-Importing own module as well as external modules Understanding Packages, Powerful Lamda function in python Programming using functions, modules and external packages.

Unit VI : Working with Data in Python: (Hours:07)

Printing on screen- Reading data from keyboard- Opening and closing file- Reading and writing files Functions Loading Data with Pandas-Numpy

Text Book:

Martin C Brown, “Python: The Complete Reference”, MCGraw Hill

Reference Books:

1. Larry Lutz, “Python for Beginners: Step-By-Step Guide to Learning Python Programming”, CreateSpace Independent Publishing Platform, First edition, ISBN-1717410588, 9781717410580, 2018
2. Nicholas Ayden, “Python Programming”, Independently Published, First Edition, ISBN- 1707051933, 9781707051939, 2019.
3. MichałJaworski, Tarek Ziadé, “Expert Python Programming”, Packt Publishing Ltd., Third Edition, ISBN-9781789808896, 2019.

== XX ==

3CS202PC/3DS202PC/3AD202PC/3KE202PC

DATA STRUCTURES

Course Pre-requisite:
Fundamentals of programming Language & Logic Building Skills

Course Objectives:

- To understand the linear and nonlinear data Structures and its memory representations.
- To perform different operations on data structures such as insertion, deletion, searching and
- traversing.
- To understand various data searching and sorting methods with its complexity.
- To introduce various techniques for representation of the data in the real world.

Course Outcomes:
On completion of the course, the students will be able to
1. Apply various linear and nonlinear data structures
2. Demonstrate operations like insertion, deletion, searching and traversing on various data structures
3. Examine the usage of various structures in approaching the problem solution.
4. Choose appropriate data structure for specified problem domain

Syllabus:

Unit I: Introduction to Data Structures: (Hours:07)
Introduction to Data structures, Data Structure Operations, Algorithmic Notation, Complexity of algorithms. String processing: storing strings, character data type, string operations, word processing, and pattern matching algorithms.

Unit II: Array& Record Structure: (Hours: 07)
Linear arrays: Memory Representation of arrays, traversing linear arrays, insertion & deletion operations, Bubble sort, Linear search and Binary search algorithms. Multi-dimensional arrays, Pointer arrays. Record structures.

Unit III: Linked lists: (Hours: 07)
Linked lists: Memory Representation of Linked List, traversing a linked list, searching a linked list. Memory allocation & garbage collection. Insertion & deletion operations on linked lists. Header linked lists, Two- way linked lists.

Unit IV: Stack & Queue: (Hours: 07)
Stacks: Sequential Memory Representation of Stack, Arithmetic expressions: Polish notation. Quick sort, Recursion, Tower of Hanoi. Queues: Sequential Memory Representation of Queue, DeQueue, Priority queues.

Unit V: Trees: (Hours: 07)
Introduction to Trees, Binary trees, Memory Representation of Binary Tree, Traversing binary trees, Header nodes, Binary Search Tree, Heap and heap sort, Path length & Huffman's algorithm.

Unit VI: Graphs & Sorting Algorithms: (Hours; 07)
Introduction to Graphs, Memory representation of graphs, Warshalls' algorithm, operations on Graphs, Breadth First Search, Depth First Search. Sorting : Insertion Sort, Selection Sort, Radix sort, Merge Sort.

Text Books:

1. Seymour Lipschutz: Data Structures, Schaum's Outline Series, McGraw-Hill, International Editions.
2. Trembley, Sorenson: An Introduction to Data Structures with Applications, McGraw Hill.

Reference Books:

1. Ellis Horowitz, Sartaj Sahni: Fundamentals of Data Structures, CBS Publications.
2. Data Structure Using C, Balagurusamy.
3. Standish: Data Structures in Java, Pearson Education.

== XX ==

3CS400EL / 3KE400EL /3DS400EL /3AD400EL

COMMUNITY ENGAGEMENT PROJECT / FIELD PROJECT

Suggested Activities for Community Engagement/ Field Project

1. Digital Empowerment & Awareness

📌 **Bridging the Digital Divide:**

- Train villagers on **mobile banking**, UPI payments, and accessing government e-services (ration card, Aadhaar updates, PM-KISAN).
- Conduct **awareness workshops** on the **Digital India** mission and help locals register for online services.
- Develop **multilingual video tutorials** on digital literacy, mobile safety, and secure online transactions.

- Provide **training on using government portals** like eNAM (National Agricultural Market) and GeM (Government e-Marketplace).

📌 **Smart Village Initiative:**

- Set up **public Wi-Fi hotspots** in collaboration with local authorities.
- Assist in **digitizing local businesses**, helping small vendors and farmers list their products online.
- Train rural students in **basic coding and website development** for local business promotion.

2. Smart Agriculture & Technology Adoption

📌 **Tech-Driven Farming Solutions:**

- Develop an **AI-based chatbot** in regional languages to assist farmers with crop-related queries.
- Install **IoT-based soil sensors** and educate farmers on precision farming techniques.
- Train farmers in **disease detection using AI-based mobile apps** for plant health monitoring.
- Introduce **solar-powered irrigation systems** to reduce dependency on grid electricity.

📌 **Sustainable Agricultural Practices:**

- Conduct **vermicomposting and organic farming workshops** to reduce chemical usage.
- Set up **bio-fertilizer production units** using locally available waste.
- Promote **agroforestry** as a long-term sustainable practice.

3. Water & Resource Management

📌 **Water Conservation & Management:**

- Conduct **rainwater harvesting awareness** campaigns.
- Develop a **low-cost water filtration system** for villages facing potable water issues.
- Implement **drip irrigation systems** for water conservation in farming.
- Map and revive **traditional water bodies** in the region.

📌 **Energy-Efficient Practices:**

- Promote the use of **solar lamps and bio-gas units** in rural households.
- Conduct workshops on **converting agricultural waste into biomass energy**.

4. E-Commerce & Rural Entrepreneurship Development

📌 **Market Linkage & Skill Development:**

- Assist farmers in selling their products on **e-commerce platforms** like Amazon Kisan Store, Flipkart Krishi, and eNAM.
- Conduct training on **branding, packaging, and direct farm-to-market sales** strategies.
- Develop an **online directory of rural artisans** to promote handicrafts through social media.

📌 **Rural Women Empowerment:**

- Train rural women in **handicraft-based digital entrepreneurship** (e.g., selling products on Etsy, Meesho).

- Organize workshops on **self-help groups (SHGs)** and micro-financing opportunities.
- Introduce women to **low-cost, home-based business ideas** like making organic soaps, candles, or eco-friendly bags.

5. Health & Hygiene Awareness

📌 **Basic Healthcare Services:**

- Conduct **free medical check-up camps** with local health professionals.
- Develop an **AI-powered mobile application** to spread verified medical information in local languages.
- Train rural women on **maternal health, child nutrition, and menstrual hygiene management**.

📌 **Sanitation & Waste Management:**

- Organize **waste segregation drives** and teach villagers about composting.
- Introduce **biodegradable waste recycling** to generate biogas and organic manure.
- Develop **low-cost water purification units** for households.

6. Education & Skill Development

📌 **STEM Awareness Programs:**

- Conduct **STEM (Science, Technology, Engineering, and Mathematics) workshops** in rural schools.
- Develop **interactive STEM kits** for students to experiment with physics and engineering principles.
- Train rural students in **3D printing and CAD software** for technical skill enhancement.

📌 **Career & Competitive Exam Guidance:**

- Offer **guidance sessions for competitive exams** like JEE, NEET, UPSC, and MPSC for rural students.
- Organize **spoken English and communication skill development programs**.
- Help students with **resume writing, online job portals, and freelancing opportunities**.

7. Cyber Security & Online Safety

📌 **Safe Digital Practices:**

- Conduct **cybersecurity awareness workshops** on recognizing fraud, scams, and phishing attacks.
- Train rural youth on **secure online transactions** and social media privacy settings.
- Provide **hands-on training on data encryption and password management**.

8. Renewable Energy & Sustainability

📌 **Eco-Friendly Energy Solutions:**

- Promote **solar-powered microgrids** for electrification in remote areas.
- Educate villagers on **alternative energy sources** like wind and hydro power.
- Develop and implement **low-cost biogas plants** for households.

📌 **Green Village Initiative:**

- Organize **tree plantation drives** in deforested areas.

- Conduct **sustainability workshops** focusing on reducing carbon footprints.

9. Waste Management & Circular Economy

📌 **Plastic-Free Village Campaign:**

- Conduct **awareness programs on plastic pollution and alternatives**.
- Organize **recycling drives and upcycling workshops** for students and villagers.
- Train women-led SHGs in **eco-friendly bag and paper product manufacturing**.

📌 **Low-Cost Housing Solutions:**

- Develop a model for **affordable, sustainable housing** using locally available materials.
- Promote the use of **mud bricks, bamboo, and recycled materials** for low-cost homes.

10. Smart Infrastructure & Community Development

📌 **Smart Village Concept:**

- Develop a **GIS-based village mapping system** for better resource allocation.
- Set up **community radio stations** to disseminate important information.
- Install **smart lighting solutions** (solar-powered street lights) for energy efficiency.

📌 **Community-Driven Innovation:**

- Encourage students to develop **innovative low-cost solutions** to local problems.
- Facilitate **co-creation workshops** where villagers and students collaborate on ideas.

Impact of These Activities

✅ **Bridges the gap between classroom learning and real-world challenges**.
✅ **Encourages students to develop AI/IoT-based solutions for rural challenges**.
✅ **Promotes sustainability and digital literacy among rural communities**.
✅ **Fosters a sense of social responsibility and entrepreneurship in students**.

This course emphasizes experiential learning through community-based projects where students will apply computing knowledge and emerging technologies to solve real-world problems faced by society. Students will actively engage with local communities, identify challenges, and develop technology-driven solutions that foster social development and digital empowerment.

Course Objectives:

- Provide students with practical exposure to how computer science and digital technologies can address societal challenges.
- Enable students to design and deploy small-scale ICT/AI/IoT-based solutions in real community contexts.
- Develop digital awareness, problem-solving, and teamwork skills through field-based activities.
- Encourage social responsibility and ethical use of technology.

Course Outcomes:

After completing the course, students will be able to:

1. Identify societal and community problems that can be addressed through computer science applications.
2. Design small-scale prototypes or awareness modules leveraging ICT, AI, IoT, or data science.
3. Demonstrate hands-on application of emerging technologies in community settings.
4. Collaborate with local stakeholders to deploy and evaluate solutions.
5. Document outcomes of field projects and reflect on experiential learning.

Details of Syllabus / Suggested Activities:

Perform 5 - 6 Practical activities (or equivalent projects) from the list should be performed.

1. **Basic Computer Literacy Sessions** – Teach elderly or underprivileged groups how to use smartphones, send SMS/WhatsApp messages, or browse the internet safely.
2. **Digital Document Assistance** – Help community members create and manage Aadhaar-linked services, PAN, ration card updates, or digital health IDs.
3. **Awareness on Cyber Frauds** – Design and distribute pamphlets/posters on avoiding phishing calls, OTP fraud, and fake links.
4. **Assist Farmers in Using Kisan Apps** – Demonstrate the use of government agriculture apps like *PM-Kisan*, *IFFCO Kisan*, or *Kisan Suvidha*.
5. **Energy Saving Awareness Drive** – Conduct a door-to-door survey and create posters on saving electricity, switching to LED bulbs, and responsible usage.
6. **Plastic-Free Awareness Campaign** – Develop posters or short plays/skits to educate the community on reducing single-use plastics.
7. **Help in Using E-Rickshaw / EV Charging Apps** – Guide local drivers or users in using digital payment methods for EV charging stations.
8. **Community Survey on Waste Management** – Collect data about how households dispose of waste and suggest better segregation methods.
9. **Health Awareness Posters** – Prepare easy-to-understand visual guides about vaccination schedules, nutrition charts, or basic hygiene practices.
10. **Promote Local Library Digitization** – Help libraries or schools catalog their books using Excel or simple databases.
11. **Assist Shopkeepers in Using QR Codes** – Help small vendors adopt UPI/QR code payments and train them in transaction safety.
12. **Create a Local Bus/Transport Timetable** – Collect schedules from local bus stations and prepare a digital or printed timetable for commuters.
13. **Assist in Online Railway / Bus Ticket Booking** – Train rural residents in booking affordable tickets safely through apps/portals.
14. **Organize an "E-Waste Awareness Camp"** – Collect old gadgets/e-waste and educate people about proper disposal/recycling.
15. **Create Safety Awareness Charts** – For road safety, emergency contacts, and first aid—distribute in schools and community centers.
16. **Promote Rainwater Harvesting Practices** – Conduct awareness sessions and create models/demos of small harvesting systems.
17. **Awareness on Social Media Usage** – Teach responsible use of Instagram, Facebook, WhatsApp to school students; warn against oversharing personal data.

18. **Assist Local NGOs** – Help them in preparing basic presentations, brochures, or email communication for their causes.
19. **Basic Typing and Resume Help** – Help unemployed youth learn typing, create email IDs, and prepare resumes using MS Word.
20. **Digital Health Awareness** – Guide villagers in using *Aarogya Setu*, *ABHA Digital Health Card*, or booking doctor appointments online.
21. **Water Conservation Awareness** – Create charts and posters with tips on saving water at household level.
22. **Awareness on Government Scholarships** – Collect and distribute information about scholarships available for students in rural areas.
23. **Assist Schools in Online Examination Practice** – Help students get familiar with mock online tests using simple quiz apps.
24. **Digital Maps for Locality** – Create simple Google Maps entries for important community landmarks (clinics, schools, ATMs).
25. **Conduct a Poster-Making or Slogan Competition** in schools on themes like *Digital India*, *Green Energy*, or *Cyber Safety*.
26. Conduct a **digital literacy workshop** in a rural/urban community: train locals on safe mobile banking, UPI payments, and accessing government portals.
27. Develop and demonstrate a **basic Android app** for local needs (e.g., crop calendar, community notice board, health reminders).
28. Create **AI/ML-based awareness tools** (e.g., chatbot for agricultural queries or student career guidance).
29. Field visit to a rural school to conduct **STEM awareness sessions** using simple coding and robotics kits.
30. Develop a **website or e-commerce page** for local artisans, SHGs, or farmers to promote their products.
31. Implement an **IoT prototype** (e.g., soil moisture sensor, smart street lighting demo) and explain its community utility.
32. Survey local cybercafés/educational institutes to study **cybersecurity practices** and conduct awareness drives on online safety.
33. Develop and present **data-driven insights** on a local issue (e.g., water usage, electricity consumption, health trends) using open datasets.
34. Assist schools/NGOs in digitizing their records using **databases or cloud storage** and provide basic training.
35. Create and distribute **digital awareness content** (videos, infographics) on health, hygiene, and safe technology practices.

And similar relevant activities /projects on the same lines: curated and approved by the concerned Authority from time to time.

Tentative Report Template for Community Engagement Project

Title Page

- Project Title:
- Student Name & Roll Number:
- Faculty Guide Name:
- Date of Submission:

---

1. Introduction
   - Brief overview of the selected activity.
   - Why this issue was chosen.
2. Objectives
   - Clear statement of objectives (What problem is being addressed?).
3. Methodology
   - Approach taken (Survey, Workshops, Development of Solutions, etc.).
   - Tools and Technologies used (if applicable).
4. Execution & Implementation
   - Step-by-step description of how the activity was conducted.
   - Challenges faced and how they were overcome.
5. Observations & Impact Assessment
   - Community response and engagement levels.
   - Quantitative & qualitative impact (e.g., number of farmers trained, improved efficiency, awareness raised).
6. Learnings & Outcomes
   - Key takeaways for students.
   - Skills developed and knowledge gained.
7. Suggestions & Future Scope
   - Recommendations for scaling up the project.
   - How technology can further enhance the solution.
8. References & Acknowledgments
   - Any research material, websites, or experts consulted.
   - Acknowledgment of community members or organizations who contributed.

Evaluation Criteria (Total: 50 Marks)

Evaluation Parameter	Marks (Int)	Marks (Ext)
Problem Identification & Understanding	05	05
Implementation & Execution	05	05
Innovation & Use of Technology	05	05
Community Engagement & Feedback	05	05
Report Quality, Presentation, Future Scope & Recommendations.	05	05

3CS203PC/3AD203PC

OBJECT ORIENTED PROGRAMMING LAB

Course Pre-requisite:
Basic Computer Programming

Course Objectives:
Design, implement, test, and debug simple programs in an object-oriented programming language.

- To develop the knowledge of object-oriented paradigm in the Java programming language.
- To evaluate classical problems using java programming.
- To develop software development skills using java programming for real world applications.

Course Outcomes :
On completion of the course, the students will be able to

1. Design, implement, test, and debug simple programs in an object-oriented programming language.
2. Interpret the basics of object-oriented design and the concepts of encapsulation, abstraction, inheritance, and polymorphism
3. Build applications in Java by applying concepts like interfaces, packages and exception handling.
4. Make use of Java concepts like API, Applets, AWT.

List of Experiments:
This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi

1. Introduction to Object Oriented Programming and installation of JDK. Write a program to print a message "Hello World…"
2. Develop a program to explain use of Operators in java.
3. Develop a Program to study and implement Looping Statements belonging to Java**.**
4. Develop a Program to study and implement Selection Statements belonging to Java**.**
5. Develop a program to study and implement some Pyramid.
6. Develop a program to demonstrate the concept of Class, Method and Object.
7. Develop a program to study and implement the concept of Method Overloading.
8. Develop a program to study and implement concept of Constructor in Java.
9. Develop a program to study and implement concept of Constructor Overloading in Java.
10. Develop a program to study and implement the Array in Java.
11. Develop a Program on various ways to accept data through keyboard(Command Line Argument)
12. Develop a program to study and implement the concept of Inheritance.
13. Develop a program to study and implement the concept of Method Overriding.
14. Develop a program to study and implement the Abstract Class.
15. Develop a program to study and implement the concept of Interface in Java.
16. Develop a program to study and implement Exception Handling Mechanism in Java.
17. Develop a program to study and implement Java I/O.
18. Develop a program to study and implement simple Applet in java.
19. Develop a program on Applet to demonstrate Graphics, Font and Color class.
20. Develop a Program on passing parameters to applets
21. Develop a Program to create GUI application without event handling using AWT controls

22. Develop a Program to create GUI application with event handling using AWT controls
23. Develop a program on Multithreading
24. Develop a Program to create GUI application with event handling using Swing controls
25. Mini Project based on content of the syllabus. (Group of 2-3 students)

== XX ==

3DS203PC/3KE203PC

PROGRAMMING METHODOLOGY USING PYTHON LAB

Course Prerequisite:
Basics of programming Language

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of Programming Methodology using Python Lab by being able to do each of the following:

- Interpret the use of procedural statements like assignments, conditional statements, loops and function calls. Learn the syntax and semantics and create the functions in Python.
- Infer the supported data structures like lists, dictionaries and tuples in Python.
- Illustrate the application of matrices and regular expressions in building the Python programs.
- Discover the use of external modules in creating excel files and navigating the file systems.
- Describe the need for Object-oriented programming concepts in Python.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:

1. Apply the Python language syntax including control statements, loops and functions.
2. Understand the core data structures like lists, dictionaries, tuples and sets in Python to store, process and sort the data.
3. Interpret the concepts of Object-oriented programming as used in Python
4. Identify the external modules for creating and writing data to excel files and inspect the file operations to navigate the file systems.

List of Experiments:
Minimum 12 Experiments based on each concept are to be performed covering the entire syllabus.
At least two experiments should be beyond syllabi based on learning of syllabi, This list is for reference only one can prepare a list according to the syllabus.

1. Write a program to understand basic Python interpreter
2. Write a program to perform String Manipulation and Functions
3. Write a program to implement Python Data structures
4. Implement Classes and Objects using Python
5. Design a program to understand Overloading in Python
6. Design a program to understand Overriding in Python

7. Implement Inheritance using Python
8. Develop a Python code for Information hiding
9. Implement Python's Modules and Packages
10. Writ a program to illustrate concept of File handling
11. Write a code to load Data with library such as Pandas-Numpy.

Text Book:
Martin C Brown, "Python: The Complete Reference", McGraw Hill

Reference Books:

1. Larry Lutz, "Python for Beginners: Step-By-Step Guide to Learning Python Programming",
   CreateSpace Independent Publishing Platform, First edition, ISBN- 1717410588, 9781717410580, 2018
2. Nicholas Ayden, "Python Programming", Independently Published, First Edition, ISBN-
   1707051933, 9781707051939, 2019.
3. Michał Jaworski, Tarek Ziadé, "Expert Python Programming", Packt Publishing Ltd., Third Edition, ISBN-9781789808896, 2019.

== XX ==

3CS204PC/3DS204PC/3AD204PC/3KE204PC

DATA STRUCTURE LAB

Course Pre-requisite:
Basics of programming Language & Logic Building Skills

Course Objectives:
- To understand the linear and nonlinear data Structures and its memory representations.
- To perform different operations on data structures such as insertion, deletion, searching and traversing.
- To understand various data searching and sorting methods with its complexity.
- To introduce various techniques for representation of the data in the real world.

Course Outcomes:
On completion of the course, the students will be able to
1. Apply various linear and nonlinear data structure.
2. Demonstrate operations like insertion, deletion, searching and traversing on various data structures
3. Examine the usage of various structures in approaching the problem solution.
4. Choose appropriate data structure for specified problem domain

List of Experiments:
This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi

1. Write a program to find out largest number from the array and also find it's location.

2. Write a program to traverse an array and find the sum and average of data elements from an array.
3. Write a Program to a) insert an element in an array b) delete an element from an array.
4. To study and execute the Linear search method
5. To study and execute the Binary Search method
6. To study and execute the Pattern matching Algorithms (Slow and Fast)
7. To study and execute Bubble sort method.
8. To study and implement various operations on singly linked list
    a. Traversing the linked list.
    b. Insert a node at the front of the linked list.
    c. Delete a last node of the linked list.
    d. Searching a Linked list.
9. To study and implement following operations on the doubly linked list.
    a. Insert a node at the front of the linked list.
    b. Insert a node at the end of the linked list.
    c. Delete a last node of the linked list.
    d. Delete a node before specified position.
10. To study and implement following operations on the circular linked list.
    a. Insert a node at the end of the linked list.
    b. Insert a node before specified position.
    c. Delete a first node of the linked list.
    d. Delete a node after specified position.
11. Understand the stack structure and execute the push, pop operation on it.
12. Understand the Queue structure and execute the insertion, deletion operation on it.
13. Formulate and demonstrate Transforming Infix Expressions to Postfix Expression using Stack.
14. Formulate and demonstrate the Evaluation of Postfix Expression using Stack.
15. To study and execute Quick sort method.
16. Understand the Tree structure and implement the Pre-order, In-order, post-order traversing operations on it.
17. Understand the concept of Recursion and write a program to calculate factorial of a number using Recursion.
18. Understand the Heap sort and implement it on given data.
19. Understand the Insertion sort and implement it on given data.
20. Understand the Selection sort and implement it on given data.
21. To study and execute Merge sort method.
22. To study and execute Radix sort method.
23. Write a Program to implement the concept of BFS algorithm.
24. Write a Program to implement the concept of DFS algorithm.

To study and execute Josephus problem.

== XX ==

3CS207EM / 3KE207EM / 3DS207EM / 3AD207EM

ENTREPRENEURSHIP DEVELOPMENT

Course Pre-requisites:

- Basic understanding of engineering problem-solving
- Interest in innovation or business development

Course Objectives:

The course aims to:

- Introduce engineering students to the fundamentals of entrepreneurship.
- Develop the ability to identify and evaluate entrepreneurial opportunities.
- Encourage creativity and innovation as key entrepreneurial traits.
- Equip students with knowledge of business planning, models, and funding.
- Emphasize the role of marketing, intellectual property, and ethical practices.

Course Outcomes:

At the end of the course, students will be able to:

1. Understand and explain key entrepreneurial concepts and the startup ecosystem.
2. Identify, assess, and evaluate business opportunities using feasibility analysis.
3. Develop a basic business plan including financial, marketing, and legal elements.
4. Recognize the importance of innovation, funding sources, and IP rights in startups.
5. Apply entrepreneurial thinking to engineering problems and real-world challenges.

Syllabus:

Unit I: Introduction to Entrepreneurship & Opportunity Identification (8 Hours)

Evolution and Importance of Entrepreneurship in the Engineering Context, Myths about Entrepreneurs, Role of Engineers as Entrepreneurs, Types of Startups, Entrepreneurial Traits, Ideation Techniques, Innovation vs. Invention, Identifying Opportunities and Gaps in the Market, Tools for Idea Generation (Brainstorming, SCAMPER, TRIZ).

Unit II: Feasibility Analysis, Business Models & Planning (7 Hours)

Feasibility Analysis: Product/Service, Market, Organizational & Financial, Business Model Canvas – Key Components, Elements of a Business Plan, Introduction to Financial Planning – Break-even, Basic Forecasting, Industry Analysis and Competitor Mapping, Value Proposition Design.

Unit III: Funding, Marketing & Intellectual Property (7 Hours)

Sources of Funding: Angel Investors, VCs, Bootstrapping, Government Schemes, Role of Incubators and Accelerators, Marketing for Startups – 4Ps, Digital Marketing Basics, Importance of IP: Patents, Copyrights, Trademarks, Trade Secrets, Ethics in Entrepreneurship, Role of Entrepreneurship in Sustainable Development.

Textbook:

Bruce R. Barringer & R. Duane Ireland, *Entrepreneurship: Successfully Launching New Ventures*, Pearson Education, 3rd Edition

Reference Books:

1. Poornima M.C., *Entrepreneurship Development – Small Business Enterprises*, Pearson
2. Ram Chandran, *Entrepreneurial Development*, Tata McGraw Hill
3. Arya Kumar, *Entrepreneurship*, Pearson
4. Khanka, S. S., *Entrepreneurial Development*, S. Chand & Co.
5. Kanishka Bedi, *Management and Entrepreneurship*, Oxford University Press
6. Badhai, B., *Entrepreneurship for Engineers*, Dhanpat Rai & Co.

==XX==

SEM IV

4CS205PC/4DS205PC/4AD205PC/4KE205PC

DATA COMMUNICATION AND NETWORKING

Course Pre-requisite:
Computer and Data Communication Requirements

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of Data Communication and Networking by being able to do each of the following:

- Study the basic taxonomy and terminology of the digital communication system & computer networking and enumerate the layers of OSI model and TCP/IP model.
- Acquire knowledge of Application layer paradigms and protocols.
- Study Transport layer design issues, Transport layer services, and protocols.
- Gain core knowledge of Network layer routing protocols and IP addressing.
- Study data link layer concepts, design issues, and protocols.

Course Outcomes (CO's):
On completion of the course, the students will be able to:

1. Describe the functions of each layer in OSI and TCP/IP model.
2. Describe the Transport layer and Transport layer services.
3. Classify the routing protocols and analyze how to assign the IP addresses for the given network.
4. Explain the functions of Application layer and Presentation layer paradigms and Protocols.
5. Describe the functions of data link layer and explain the protocols.
6. Explain the types of transmission media with real time applications.

Syllabus:

Unit I: Introduction to Network Models (Hours: 07)
Data Communication Components: Basic network types, switching mechanisms, Internet standards. **Layered Architecture**: Overview of the OSI and TCP/IP models, focusing on key functions of each layer.

Unit II: (Hours: 07)
Transmission Media: Guided vs. Unguided media, key characteristics, and real-time application examples. **Switching Techniques**: Circuit, Packet, and Virtual Circuit Switching.

Unit III: Application Layer Protocols - (Hours: 07)
Application Layer Overview: Client-server model, APIs, P2P networking. **Key Protocols**: HTTP, FTP, SMTP, DNS, focusing on their importance in internet communication.

Unit IV: Transport Layer Fundamentals - (Hours: 07)
Transport Layer Overview: Principles of connectionless and connection-oriented services. **Key Protocols**: UDP, TCP – Concepts of reliability, congestion control, flow control. **Multiplexing and Demultiplexing**: Basic mechanisms and their relevance in data transmission.

Unit V: Network Layer Protocols and IP Addressing - (Hours: 07)
Network Layer Overview: Services provided by the network layer, datagram vs. virtual circuit approaches. **IP Addressing**: IPv4, IPv6, NAT, DHCP, and ICMP functionalities. **Routing**: Introduction to routing algorithms (Distance Vector and Link-State), basic concept of forwarding.

Unit VI: Data Link Layer and MAC Protocols - (Hours: 07)
Data Link Layer Services: Framing, error detection (CRC, Checksum), and correction techniques. **MAC Layer**: Concepts of LAN addressing, ARP, CSMA/CD, and PPP. **Link Layer Protocols**: HDLC and Point-to-Point Protocol.

Text Books:
1. Behrouz A. Forouzan: Data Communication and Networking, (5/e) (TMH)
2. James F. Kurose & K W Ross: Computer Networking, Pearson Education (LPE)

Reference Books:
1. William Stallings: Data & Computer Communications, 6/e, Pearson Education
2. William L. Schweber: Data Communication, McGraw Hill
3. Douglas E. Comer: Computer Network & Internet, Addison Wesley.
4. Andrew S. Tanenbaum: Computer Networks, PHI (5E)
5. Leon Garcia & Widjaja: Communication Networks, TMH

== XX ==

4CS206PC/4DS206PC/4AD206PC/4KE206PC

OPERATING SYSTEM

Course Pre-requisite:
Discrete Structures, Data Structure, Any programming Language

Course Objectives:
1. To make students aware of the kernel and shell structure of the operating systems.
2. To make students aware of the purpose, structure and functions of operating systems
3. To equip students with understanding of the various scheduling algorithms in OS.
4. To make students aware of understanding of memory management in different OS.

Course Outcomes: (Expected Outcome):
On completion of the course, the students will be able to:

- Explain memory management issues like external fragmentation, internal fragmentation.
- Illustrate multithreading and its significance.
- List various protection and security mechanisms of OS.
- Analyze and solve the scheduling algorithms.
- Analyze the deadlock situation and resolve it.
- Compare various types of operating systems

Syllabus:

Unit I: Introduction to OS (Hours: 07)
Introduction: Operating System definition, OS Evolution, Components and Services, Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Interprocess Communication, Threads Overview, Multithreading Models, Threading Issues, Java Threads

Unit II: Process Scheduling (Hours: 07)
Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR, Priority, Multilevel Queue, Multilevel Feedback Queue Scheduling

Unit III: Process Synchronization (Hours: 07)
Process Synchronization Basics: The Critical-Section Problem, Synchronization Hardware, Semaphores, Monitors, Deadlocks: Definition & Characterization, Deadlocks Prevention, Avoidance, Detection and Recovery from Deadlock

Unit IV: Memory Management (Hours: 07)
Memory Management Background, Swapping, Contiguous Memory Allocation Schemes, Paging, Segmentation, Virtual Memory Management: Background, Demand paging scheme, Process Creation, Page Replacement Policies, Allocation of Frames, Thrashing

Unit V: File System (Hours: 07)
File-System Interface; Directory Structure, File-System Mounting, File Sharing &Protection, File- System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free- Space Management.

Unit VI: I/O System (Hours:07)
I/O Systems: Overview, I/O Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O to Hardware Operations , Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure.

Text Book :
Avi Silberschatz, P.B.Galvin, G.Gagne: “Operating System Concepts” (9/e) John-Wiley & Sons.

Reference Books:

1. A.S.Tanenbaum “Modern Operating Systems” Pearson Education.
2. William Stallings “Operating Systems” Prentice-Hall.
3. D. M. Dhamdhere “Operating Systems” Tata McGraw-Hill.
4. P. Balkrishna Prasad: “Operating Systems” Scitech Publications (I) Pvt. Ltd.

== XX ==

4CS207PC/4KE207PC

THEORY OF COMPUTATION

Course Pre-requisites

- Discrete Mathematics
- Data Structures

Course Objectives:

- To understand automata theory and its operation.
- To understand mathematical expressions for formal languages.
- To learn implementing the state machines for solving real life problems.
- To study computing machines and comparing different types of computational models.
- To understand the fundamentals of problem decidability and Un-Decidability.

Course Outcomes: (Expected Outcome):

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

1. To construct finite state machines to solve problems in computing.
2. To write regular expressions for the formal languages.
3. To construct and apply well defined rules for parsing techniques in compiler.
4. To construct and analyze Push Down, Turing Machine for formal languages
5. To express the understanding of the Chomsky Hierarchy.
6. To express the understanding of the decidability and un-decidability problems.

Syllabus:

Unit I: Finite State Machines: (Hours: 07)

Alphabet, String, Formal and Natural Language, Operations, Definition and Design DFA (Deterministic Finite Automata), NFA (Non-Deterministic Finite Automata), Equivalence of NFA and DFA: Conversion of NFA into DFA, Conversion of NFA with epsilon moves to NFA, Minimization Of DFA, Definition and Construction of Moore and Mealy Machines, Inter conversion between Moore and Mealy Machines. Minimization of Finite Automata. (Construction of Minimum Automaton).

Unit II: Regular Expression and Regular Grammar: (Hours: 07)

Definition and Identities of Regular Expressions, Construction of Regular Expression of the given Language, Construction of Language from the RE, Conversion of FA to RE using Arden’s Theorem, Inter-conversion RE to FA, Pumping Lemma for RL, Closure properties of RLs (proofs not required), Regular grammar, Equivalence of RG (RLG and LLG) and FA.

Unit III: Context Free Grammar and Languages: (Hours: 07)
Introduction, Formal Definition of Grammar, Notations, Derivation Process: Leftmost Derivation, Rightmost Derivation, Derivation Trees, Construction of Context-Free Grammars and Languages, Pumping Lemma for CFL, Simplification of CFG, Normal Forms (CNF and GNF), Chomsky Hierarchy.

Unit IV: Pushdown Automata: (Hours: 07)
Introduction and Definition of PDA, Construction of PDA, Acceptance of CFL, Equivalence of CFL and PDA: Inter-conversion, Introduction of DCFL and DPDA, Enumeration of properties of CFL, Context Sensitive Language, Linear Bounded Automata.

Unit V: Turing Machines: (Hours: 07)
Formal definition of a Turing Machine, Design of TM, Computable Functions, Church's hypothesis, Counter machine, Variants of Turing Machines: Multi-tape Turing machines, Universal Turing Machine.

Unit VI: Decidability and Un-Decidability: (Hours: 07)
Decidability of Problems, Halting Problem of TM, Un-Decidability: Recursive enumerable language, Properties of recursive & non-recursive enumerable languages, Post Correspondence Problem, Introduction to Recursive Function Theory.

Text Book

1. Hopcraft H.E. & Ullman J: Introduction to Automata Theory, Languages and Computation
2. Peter Linz: An Introduction to Formal Languages and Automata

Reference Book

1. Rajesh K. Shukla: Theory of Computation, CENGAGE Learning, 2009.
2. K V N Sunitha and N Kalyani: Formal Languages and Automata Theory, McGraw Hill,2010
3. Lewis H.P. and Papadimition C.H.: Elements of Theory of Computation
4. Mishra &Chandrashekharan: Theory of Computation
5. C. K. Nagpal: Formal Languages and Automata Theory, Oxford University Press, 2011.
6. Vivek Kulkarni : Theory of Computation, OUP India, 2013

== XX ==

4DS207PC/4AD207PC

PROBABILITY AND STATISTICS FOR DATA SCIENCE

Course Objective:
- Understand and apply probability concept.
- Learn foundational concepts and applications of probability theory and common probability distributions.

- Understand and apply measures of central tendency, dispersion, correlation, and regression analysis in statistical analysis.
- Compute and interpret the correlation coefficient.
- Understand the computational details behind certain numerical methods.
- Gain proficiency in interpolation methods and numerical techniques for solving first-order differential equations.

Course Outcomes: (Expected Outcome):

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

1. Understand about the collection of the data, condensation, and summarization into a compact form.
2. Apply probability axioms, analyze distributions, calculate conditional probabilities, and understand the central limit theorem.
3. Understand and compute measures of central tendency, dispersion, correlation, and regression for data analysis.
4. Apply numerical methods for curve fitting using least squares, including fitting straight lines and higher-degree curves.
5. Utilize numerical methods such as Newton-Raphson, false position, Gauss elimination, Gauss-Jordan, Gauss-Seidel, and matrix operations for solving equations and eigenvalue computations.
6. Apply interpolation techniques and numerical methods to solve differential equations accurately and efficiently.

Syllabus:

Unit: I (Hours: 07)

Meaning of experiment, random experiment, deterministic and non-deterministic models. Definition of the term: Outcome, sample space (finite and infinite), discrete sample space, Elementary event, Compound event, Complementary event, Favorable event, Equality-likely events, Sure event, Impossible event. Concept of Occurrence of an event, Union, and Intersection of two or more events, Exhaustive events, mutually exclusive events, Representation of sample space and events by Venn diagram, Occurrence of (i) at least one of the given events (ii) all of the events (iii) none of the given events, Example and problem.

Unit: II (Hours: 07)

Axioms of Probability, centre limit theorem, conditional probability, Bayes' rule, Bernoulli trials, probability mass function, continuous random variable, probability density function, probability distributions: Binomial distribution, Poisson distribution, normal distribution.

Unit: III (Hours: 07)

Measure of central tendency: Moments, Expectation, dispersion, skewness, kurtosis, expected value of two-dimensional random variable, Linear Correlation, correlation coefficient, rank correlation coefficient, Regression, Line of Regression.

Unit: IV (Hours: 07)
Curve fitting by the numerical method: Curve fitting by of method of least squares, fitting of straight lines, second degree parabola and more general curves.

Unit: V (Hours: 07)
Newton Raphson method, false position method, Gauss elimination, Gauss Jordan, Gauss Seidel, Matrix Inversion by Gauss Jordan method, Eigen value of a matrix by power method.

Unit: VI (Hours: 07)
Interpolation: Forward Difference, Backward Difference, Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Lagrange's Interpolation. Numerical Solution of First order differential equation: Picard's method, Taylor series, Euler's Method, Modified Euler's Method, Runge-Kutta Method of fourth order.

Textbook:

1. Agarwal, B. I. (2047). Programmed Statistics, Third Edition, New Age International Publisher, New Delhi.
2. S. S. Shastri. Introductory Methods of Numerical Methods, PHI, Vol 2.

Reference Book:

1. Gupta S. C. and Kapoor V. K. (2019) Fundamental of Mathematical Statistics.
2. Numerical Methods for Scientifi c and Engineering Computation' by M. K. Jain, S. R. K. Iyengar, R. K. Jain
3. Myer,P. I. (1970): Introductory Probability and Statistical Application, Oxford & IBH Publishing, New Delhi.

== XX ==

4CS208PC/4DS208PC/4AD208PC/4KE208PC

DATA COMMUNICATION & NETWORKING LAB

Course Pre-requisite:
Computer and Data Communication Requirements

Course Objectives:
- To understand the working principle of various communication protocols
- To understand and analyze the signal flow in a digital communication system.
- To analyze error performance of a digital communication system in presence of noise and other interferences.
- To evaluate the errors using various error detection & correction techniques.
- To understand network based protocols in data communication and networking.

Course Outcomes :
On completion of the course, the students will be able to
1. Analyze performance of various communication protocols
2. Implement Configure various network protocols.
3. Compare IP Address classes of networks

List of Experiments:
This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi

1. To study various LAN topologies and their creation using network devices, cables and computers.
2. To connect the computers in Local Area Network.
3. Familiarization with Networking Components and devices: LAN Adapters, Hubs, Switches, Routers etc.
4. Write a program of bit stuffing used by Data Link Layer
5. Write a program to implement CRC(Cyclic Redundancy Check)
6. Write a program to implement Checksum
7. Write a program to implement Sliding window
8. Configure Internet connection and use IP-Config, PING / Tracer and Net stat utilities to debug the network issues.
9. Configuration of TCP/IP Protocols in Windows and Linux.
10. Transfer files between systems in LAN using FTP Configuration, install Print server in a LA and share the printer in a network.
11. Write a C Program to determine if the IP Address is in Class A, B, C, D, or E
12. Write a C Program to translate Dotted Decimal IP Address into 32 Bit Address.
13. Configure Host IP, Subnet Mask and Default Gateway in a System in LAN(TCP/IP Configuration)

== XX ==

4CS209PC/4DS209PC/4AD209PC/4KE209PC

OPERATING SYSTEM LAB

Course Pre-requisite:
Basic computer programming

Course Objectives:
- To make students aware of the kernel and shell structure of the operating systems.
- To make students aware of the purpose, structure and functions of operating systems
- To equip students with understanding of the various scheduling algorithms in OS.
- To make students aware of understanding of memory management in different OS.

Course Outcomes:
On completion of the course, the students will be able to
1. Explain memory management issues like external fragmentation, internal fragmentation.
2. Illustrate multithreading and its significance.
3. List various protection and security mechanisms of OS.
4. Analyze and solve the scheduling algorithms.
5. Analyze the deadlock situation and resolve it.
6. Compare various types of operating systems

List of Experiments:
This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi

1. To study Linux Operating System along with its installation.

2. To Study and Execute basic file commands and process related open source Ubuntu commands
    a. Commands to view all executing, block and suspended process.
    b. Command to check and change the priority of process CPU utilization for executing processes.
    c. Commands to check for child process, sub-processes, process tree, abort & end process and all other basics commands related to processes
3. Write a program for multithreading using C.
4. To simulate First Come First Serve & Shortest Job First process scheduling algorithm
5. To simulate Shortest Job First process scheduling algorithm
6. To simulate Pre-emptive Shortest Job First process scheduling algorithm
7. To implement Round Robin Process scheduling Algorithm
8. To implement Priority Based Process scheduling Algorithm
9. To implement and analyze multi-level queue scheduling algorithm
10. To implement the following file allocation strategies.
11. To simulate paging technique of memory management.
12. To implement the FIFO page replacement policy
13. To implement the LRU page replacement policy
14. To implement the optimal page replacement policy
15. To simulate producer-consumer problem using semaphores.
16. To implement Dining-Philosophers problem to deal with concurrency control mechanism.
17. To implement contiguous memory allocation strategies to detect fragmentation using: First Fit, Best Fit and Worst Fit.
18. To implement FCFS Disk Scheduling algorithm
19. To implement SCAN Disk Scheduling algorithm
20. To implement C-SCAN Disk Scheduling algorithm
21. To simulate Banker's algorithm for deadlock avoidance
22. To implement following memory management techniques
23. Implement MVT and MFT where memory block size is 100 for 5 processes. Enter no. of blocks for each process and calculate internal fragmentation.
24. To simulate LFU page replacement algorithms
25. To simulate the Single level directory file organization techniques.
26. To Simulate bankers algorithm for Dead Lock Avoidance (Banker's Algorithm)

== XX ==

4CS306VS/4DS306VS/4AD306VS/4KE306VS

Computing Skill #1 LAB (VSEC-III)

(Based on technology like -Python/Django etc. to be decided by Individual Dept. of respective College)

Course Prerequisite:
Basic knowledge of any Programming Language

Course Objectives:

- To be able to program design with functions using Python.
- To understand data and information processing techniques.
- To understand to Design a program to solve the problems.
- To be able to access database using python programming.
- To be able to design web applications using python programming.

Course Outcomes :

On completion of the course, the students will be able to

1. Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python
2. Interpret different Decision Making statements, Functions, Object oriented programming in Python
3. Summarize different File handling operations
4. Explain how to design GUI Applications in Python and evaluate different database operations
5. Develop applications using Django framework or Flask

List of Experiments:

This is a sample list of Experiments, **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi

1. Write python program to store data in list and then try to print them.
2. Write python program to print list of numbers using range and for loop
3. Write python program to store strings in list and then print them.
4. Write python program in which an function is defined and calling that function prints Hello World.
5. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST
6. 2017”
7. Write a program to create, append, and remove lists in python.
8. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
9. Write a program to demonstrate working with tuples in python.
10. Write a program to demonstrate working with dictionaries in python.
11. Write a python program to find largest of three numbers.
12. Write python program in which an function(with single string parameter) is defined and calling that function prints the string parameters given to function.
13. Write python program in which an class is define, then create object of that class and call simple print function define in class.
14. Write a Python script that prints prime numbers less than 20.
15. Write a python program to find factorial of a number using Recursion.
16. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
17. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
18. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
19. Write a Python class to convert an integer to a roman numeral.
20. Write a Python class to implement pow(x, n)
21. Write a Python class to reverse a string word by word.
22. Accessing and working with databases using Python.
23. Create data frame from .csv files and operations on it.
24. Plotting various graphs using Python.
25. Developing basic GUI using Python.
26. Developing web applications using Django framework or Flask

Reference Books :

1. “Core Python Programming”, R. NageswaraRao, dreamtech press.

2. “Python Programming A Modular Approach With Graphics, Database, Mobile and WebApplications”, SheetalTaneja, Naveen Kumar, Pearson.
3. Python Web Development with Django By Jeff Forcier, Paul Bissex, Wesley J Chun, Addison-Wesley Professional.
4. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning
5. Allen B. Downey , “ Think Python: How to Think Like a Computer Scientist”, Second Edition,
6. Shroff/O’Reilly Publishers
7. John V Guttag. “Introduction to Computation and Programming Using Python”, Prentice Hall of India
8. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, “Data Structures and Algorithms in Python”, Wiley
9. Introduction to Computation and Programming using Python, by John Guttag, PHI Publisher, Revised and Expanded version (Referred by MIT)

== XX ==

4CS217EM / 4KE217EM / 4DS217EM /4AD217EM

SOCIAL SCIENCES & ENGINEERING ECONOMICS

Course Objectives:

By the end of the course, students will:

- Understand the importance of social sciences and economics in engineering.
- Gain knowledge of the Indian Constitution, governance, and policy-making.
- Learn about economic principles, market trends, and financial institutions.
- Examine the impact of science, technology, and economic transformations on society.
- Understand professional ethics and responsibilities with socioeconomic considerations.

Course Outcomes (COs):

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

1. Understand the significance of social sciences and economic principles in engineering.
2. Analyze the role of governance, laws, and policies in shaping society and business environments.
3. Apply economic and market principles to assess financial systems and business trends.

Syllabus:

Unit I: Social Sciences and Governance (7 Hours) [CO1]

Importance of Social Sciences in Engineering, Indian Constitution: Salient features, Fundamental Rights and Duties, Directive Principles of State Policy, Governance in India: Structure of Indian Parliament, Role and Powers of the President, Prime Minister, and Council of Ministers.

Unit II: Society, Science, and Technology (7 Hours) [CO2]

Impact of Science and Technology on Culture and Civilization, Human Society: Structure, community groups, social institutions, Marriage and Family Systems: Types, functions, and modern challenges, Production & Business Organizations: Factors of production, laws of return, types of business entities.

Unit III: Economics and Market Trends (8 Hours) [CO3]

Nature and Scope of Economics: Special significance to engineers, Economic Development: Characteristics of underdevelopment, obstacles to economic growth, vicious circle of poverty. Banking & Financial Systems: Functions of Central and Commercial Banks, GST overview, Market Structures: Perfect & imperfect competition, monopoly, pricing mechanisms.

Textbook:

1. Pylee M. V. : Constitutional Govt. in India, S. Chand and Co.
2. C N Shankar Rao: Sociology, S. Chand and Co.

Reference Books:

1. Dewett and Varma J.D. : Elementary Economic Theory, S. Chand and Co.
2. A. N. Agrawal : Indian Economy, Problem of Development and Planning (Wiley Eastern Ltd), New Delhi.
3. S. K. Mishra : Indian Economy, Its Development Experience. Himalaya Pub. House, Bombay.
4. E. Kuper : Economics of W.R. Development, McGraw Hill Co.,
5. Brij Kishore Sharma. : The Constitution of India, PHI.
6. Mahajan : The Constitution of India, S. Chand, New Delhi.
7. Maclaver and Page : Principle of Sociology.
8. Davis K. : Human Society
9. Datt R.K. : Indian Economy, S. Chand and Comp. New Delhi P. M. Sundharam
10. Dhingra I.C. : Indian Economy

==XXX==