Computer Science Syllabus Grid 2025-26

SEMESTER	SUBJECT
I	Computer Fundamentals and Office Automation(T) Computer Fundamentals and Office Automation(P) Problem Solving using C (T) Problem Solving using C (P)
II	Data Structures using C (T) Data Structures using C (P) Digital Logic Design(T) Digital Logic Design(P)
III	Object Oriented Programming using Java (T)(Major) Object Oriented Programming using Java (P) Data Structures using C (T) Data Structures using C (P) Computer Organization (T) Computer Organization (P) Operating Systems (T) Operating Systems (P) Object Oriented Programming using Java (T)(Minor) Object Oriented Programming using Java (P)
IV	Database Management System (T)(Major) Database Management System (P) Object Oriented Software Engineering (T)(Major) Object Oriented Software Engineering (P) Data Communications and Computer Networks(T) Data Communications and Computer Networks(P) Database Management System (T)(Minor) Database Management System (P) Object Oriented Software Engineering (T)(Minor) Object Oriented Software Engineering (P)
V	Web Interface Designing Technologies (T) Web Interface Designing Technologies (P) Web Applications Development using PHP & MYSQL (T)(Major) Web Applications Development using PHP & MYSQL (P) Foundations of Data Science (T) Foundations of Data Science -- (P) Application development using Python (T) Application development using Python (P) Web Applications Development using PHP & MYSQL (T)(Minor) Web Applications Development using PHP & MYSQL (P) Internet of Things (T) (Minor) Internet of Things (P)
VI	Semester Internship

ST. JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

I SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma1-1601 COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Max.Marks:100

Course Objectives:

1. **Understand foundational computing concepts**, including number systems, the evolution of computers, block diagrams, and generational progress.
2. **Develop knowledge of computer architecture**, focusing on system organization and networking fundamentals.
3. **Acquire practical skills in document creation**, formatting, and digital presentations using word processing tools.
4. **Gain proficiency in spreadsheet operations**, such as data entry, formulas, functions, and charting techniques.
5. **Introduce data visualization and basic modelling principles**, fostering analytical thinking in structuring and interpreting data sets.

Course Outcomes:

1. At the End of the Course, The Students will be able to **Explain** different number systems, the historical evolution of computers, and identify key components in a block diagram.

2. Learners will demonstrate **basic blocks of a computer and fundamental networking knowledge**.

3. Learners will create professional-level documents and **design visually appealing presentations** using word processing software and presentation software.

4. Learners will manipulate data within spreadsheets, apply formulas, and **generate accurate summaries and visualizations**.

5. Learners will apply data modelling techniques to **analyze, organize, and represent data effectively** in various scenarios**.**

Unit 1. Number Systems, Evolution, Block Diagram and Generations:

Number Systems: Binary, Decimal, Octal, Hexadecimal; conversions between number systems.

Evolution of Computers: History from early mechanical devices to modern-day systems.

Block Diagram of a Computer: Components like Input Unit, Output Unit, Memory, CPU (ALU + CU). **Generations of Computers**: First to Fifth Generation – technologies, characteristics, examples.

Unit 2. Basic organization and N/W fundamentals:

Computer Organization: Functional components – Input/Output devices, Storage types, Memory hierarchy.

Types of Computers: Micro, Mini, Mainframe, and Supercomputers.

Networking Fundamentals: Definition, need for networks, types (LAN, WAN, MAN), topology (Star, Ring, Bus).

Internet Basics: IP Address, Domain Name, Web Browser, Email, WWW.

Unit 3. Word Processing and presentations:

Word Processing Basics: Using MS Word/Google Docs – formatting, styles, tables, mail merge.

Presentation Tools: Using PowerPoint/Google Slides – slide design, animations, transitions.

Applications: Creating resumes, reports, brochures, and presentations.

Keyboard Shortcuts

Unit 4. Spreadsheet Basics:

Spreadsheet Concepts: Understanding rows, columns, cells in tools like MS Excel/Google Sheets, cell referencing.

Functions and Formulae: SUM, AVERAGE, IF, COUNT.

Charts and Graphs: Creating visual representations

Data Handling: Sorting, filtering, conditional formatting.

Text Functions: LEFT, RIGHT, MID, LEN, TRIM, CONCAT, TEXTJOIN

Advanced Functions: Logical: IF, AND, OR, IFERROR, **Lookup:** VLOOKUP, HLOOKUP, XLOOKUP, INDEX, MATCH

Unit 5. Data Analysis and Visualization:

Conditional Formatting: Custom rules, Color scales, Icon sets, Data bars

Data Analysis Tools: Pivot Tables and Pivot Charts, Data Validation (Drop-downs, Input Messages, Error Alerts), What-If Analysis: Goal Seek, Scenario Manager, Data Tables

Charts and Dashboards: Creating Interactive Dashboards, Using slicers with Pivot Tables, Combo Charts and Sparklines

Productivity Tips: Using Named Ranges, Freeze Panes, Split View

Textbooks:

1. Fundamentals of Computers, Reema Thareja, Oxford University Press, Second Edition
2. Fundamentals of Computers, V. Rajaraman – PHI Learning
3. Introduction to Computers by Peter Norton – McGraw Hill
4. Microsoft Office 365 In Practice by Randy Nordell – McGraw Hill Education

References:

1. Excel 2021 Bible by Michael Alexander, Richard Kusleika – Wiley
2. Networking All-in-One For Dummies by Doug Lowe – Wiley
3. Microsoft Official Docs and Training: https://learn.microsoft.com
4. Google Workspace Learning Center: https://support.google.com/a/users/

Activities:

Outcome: At the End of the Course, The Students will be able to explain different number systems, the historical evolution of computers, and identify key components in a block diagram.

Activity: Create a digital poster or infographic comparing number systems (binary, decimal, octal, hexadecimal) and illustrating the timeline of computer generations with key innovations.

Evaluation Method: Rubric-based assessment of the poster presentation on a 10-point scale focusing on:

- Accuracy of number system conversions
- Visual organization and creativity
- Correct identification of block diagram components

Outcome: Learners will demonstrate basic blocks of a computer and fundamental networking knowledge.

Activity: Design a concept map showing the internal architecture of a computer and types of networks (LAN, WAN, MAN), including devices and topologies.

Evaluation Method: Checklist-based peer review and instructor validation:

- Completeness of the map
- Correctness of networking concepts
- Use of appropriate terminology
- Logical flow and structure of the map

Outcome: Learners will create professional-level documents and design visually appealing presentations using word processing software and presentation software.

Activity: Prepare a formal report (e.g., project proposal) in a word processor and present it using a slide deck with transitions, embedded media, and design elements.

Evaluation Method: Performance-based evaluation using a 10-point scoring scale:

- Formatting and structure of the document
- Presentation aesthetics and clarity
- Communication skills during presentation

Outcome: Learners will manipulate data within spreadsheets, apply formulas, and generate accurate summaries and visualizations.

Activity: Analyze a dataset (e.g., student scores or sales data) using spreadsheet software. Apply formulas (SUM, AVERAGE, IF, VLOOKUP) and create relevant charts.

Evaluation Method: Practical test with a rubric:

- Correct use of formulas
- Accuracy of data summaries

Outcome: Learners will apply data modelling techniques to analyze, organize, and represent data effectively in various scenarios.

Activity: Prepare an interactive dashboard for a given data set using EXCEL.

Evaluation Method: Evaluation of the dashboard on a 10-point scoring scale:

- Presentation aesthetics and clarity
- Interactiveness
- Communication skills during presentation

****

ST. JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

I SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma1-1651 COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION LAB

Max.Marks:50

Course Objectives:

1. **Acquire practical skills in document creation**, formatting, and digital presentations using word processing tools.
2. **Gain proficiency in spreadsheet operations**, such as data entry, formulas, functions, and charting techniques.
3. **Introduce data visualization and basic modelling principles**, fostering analytical thinking in structuring and interpreting data sets.

Course Outcomes:

At the End of the Course, The Students will be able to:

1. Create professional-level documents and **design visually appealing presentations** using word processing software and presentation software.
2. Manipulate data within spreadsheets, apply formulas, and **generate accurate summaries and visualizations**.
3. Apply data modelling techniques to **analyze, organize, and represent data effectively** in various scenarios**.**

List of Experiments:

1. Demonstration of Assembling and Dessembling of Computer Systems.
2. Identify and prepare notes on the type of Network topology of your institution.
3. Prepare your resume in Word.
4. Using Word, write a letter to your higher official seeking 10-days leave.
5. Prepare a presentation that contains text, audio and video.
6. Using a spreadsheet, prepare your class Time Table.
7. Using a Spreadsheet, calculate the Gross and Net salary of employees(Min 5) considering all the allowances.
8. Generate the class-wise and subject-wise results for a class of 20 students. Also generate the highest and lowest marks in each subject.
9. Using IF, AND, OR, and IFERROR to Automate Grade Evaluation.
    a. Create a table of student scores in different subjects.
    b. Use IF to assign grades (A/B/C/Fail).
    c. Use IFERROR to handle missing scores or invalid data.
10. Employee Database Search Using VLOOKUP, HLOOKUP, XLOOKUP, INDEX, and MATCH
    a. Create a database of employees (Name, ID, Department, Salary).
    b. Implement VLOOKUP to search by employee ID.
    c. Use HLOOKUP to extract department heads by role.
    d. Apply XLOOKUP for more flexible searches.

e. Use INDEX + MATCH as an alternative to VLOOKUP.

11. Sales Report Analysis Using Pivot Tables and Charts
    a. Use a dataset of product sales (Product, Region, Date, Quantity, Revenue).
    b. Create Pivot Tables to summarize data by region/product.
    c. Insert Pivot Charts for visual analysis (e.g., bar, line).
    d. Add slicers to make the dashboard interactive.
12. Designing a Data Entry Form with Drop-downs and Input Rules
    a. Create a student registration form.
    b. Add drop-down lists for course selection using Data Validation.
    c. Add input messages to guide users.
    d. Add error alerts for wrong entries.
13. Monthly Budget Planning using Goal Seek and Scenario Manager
    a. Create a simple personal budget (income, expenses, savings).
    b. Use Goal Seek to determine income needed to save a desired amount.
    c. Use Scenario Manager to compare different budgeting scenarios (best/ worst/ realistic case).
    d. Create a one-variable Data Table to analyze how different expenses affect savings.
14. Dashboard Creation Using Combo Charts, Sparklines & Slicers
    a. Use existing sales or attendance data.
    b. Insert combo charts (e.g., column + line).
    c. Add sparklines to show trends.
    d. Use slicers with Pivot Tables to control dashboard elements.
    e. Finalize and format for interactivity.

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

I SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma2-1601 **PROBLEM SOLVING USING C** Max.Marks:100

Course Objectives:

1. Understand the fundamentals of computer programming, Apply structured problem solving approaches using algorithms, flowcharts, and C programming constructs.
2. Develop efficient logic using decision-making, loop, and jump control statements.
3. Utilize derived data types like arrays and strings for modular program design.
4. Design and implement modular solutions using functions, recursive logic, pointer operations, and dynamic memory management.
5. Handle complex data structures including structures, unions, and text file operations.

Course Outcomes:

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

1. Understand basic computing concepts, programming paradigms and write structured C programs.
2. Apply control flow statements to solve logical and repetitive tasks in C.
3. Implement arrays and string operations to manage and manipulate data efficiently.
4. Design modular code using functions, recursion, and appropriate parameter passing.
5. Utilize pointers and memory operations for effective data handling. Demonstrate competence in dynamic memory allocation and text file processing.

Unit 1. Introduction to computer programming:

Introduction, Types of software, Compiler and interpreter, Concepts of Machine level, Assembly level and high-level programming, Flowcharts and Algorithms,
Fundamentals of C: History of C, Features of C, C Tokens-variables and keywords and identifiers, constants and Data types, Rules for constructing variable names, Operators, Structure of C program, Input /output statements in C - Formatted and Unformatted I/O

Unit 2. Control statements:

Decision making statements: if, if else, else if ladder, switch statements.
Loop control statements: while loop, for loop and do-while loop. Jump Control statements: break,continue and goto.

Unit 3. Derived data types in C:

Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays -Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions, Character handling functions

Unit 4. Functions:

Pointers: Pointer data type, Pointer declaration, initialization, accessing values using pointers. Pointer arithmetic, Pointers and arrays.
Function Prototype, definition and calling. Return statement. Nesting of functions. Categories of functions. Recursion (Basic Concept only). Parameter Passing by address & by value. Local and Global variables.
Storage classes: automatic, external, static and register.

Unit 5. Dynamic Memory Management:

Introduction, Functions-malloc, calloc, realloc, free
Structures: Basics of structure, structure members, accessing structure members, nested structures, array of structures, structure and functions, structures and pointers.

Unions - Union definition; difference between Structures and Unions.

Working with text files - modes: opening, reading, writing and closing text files.

Text Books:

1. Programming in ANSI C, E. Balagurusamy, Tata McGraw Hill, 6 th Edn,
2. Computer fundamentals and programming in C, Reema Theraja, Oxford University Press

Reference Books:

1. Let us C, Y Kanetkar, BPB publications
2. Head First C: A Brain-Friendly Guide, David Griffiths, Dawn Griffiths

Activities:

Outcome: Understand basic computing concepts, programming paradigms and write structured C programs.

Activity: Create a concept map of computing fundamentals and programming paradigms
(procedural, structured, object-oriented). Then, they write a structured C program (e.g., a
calculator or student grade system) using proper syntax, indentation, and modular design.

Evaluation Method: Rubric-based Code Review & Viva to check the

- ○ The correctness of the concept map
- ○ Correct use of structure (main + functions)
- ○ Identification of paradigm used
- ○ Code readability and documentation

Outcome: Apply control flow statements to solve logical and repetitive tasks in C.

Activity: Implement a program that solves a logic puzzle (e.g., number guessing game, pattern generation, or prime number finder) using if, switch, for, while, and do-while.

Evaluation Method: Automated Test Cases + Peer Review to check the

- ○ Correct use of control statements
- ○ Logical correctness of output
- ○ Efficiency and edge case handling
- ○ Peer feedback on clarity and logic

Outcome: Implement arrays and string operations to manage and manipulate data efficiently.

Activity: Build a program that stores and arranges student marks in ascending and descending order using arrays and performs string operations like concatenation, comparing, and formatting names.

Evaluation Method: Functional Demonstration + Code Walkthrough to check the

- ○ Correct array and string usage

○ Memory efficiency

○ Handling of invalid inputs

○ Explanation of sorting/searching logic

Activity:

● Recursive Problem Solver

Students write a modular program to solve a recursive problem (e.g., factorial, Fibonacci,

or Tower of Hanoi) using functions with parameters and return values.

Evaluation Method:

● Code Trace + Written Quiz

○ Correct function decomposition

○ Proper parameter passing (by value/reference)

○ Recursion depth and base case handling

○ Quiz on tracing recursive calls

Outcome: Utilize pointers and memory operations for effective data handling. Demonstrate competence in dynamic memory allocation and text file processing.

Activity: Create a program that dynamically stores user input (e.g., survey responses) using pointers and writes/reads the data to/from a text file.

Evaluation Method: Memory Debugging + File I/O Assessment to check the

○ Proper use of malloc, calloc, realloc, and free

○ Pointer arithmetic and dereferencing

○ File creation, reading, writing, and error handling

****

ST. JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

I SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma2-1601 PROBLEM SOLVING USING C LAB Max.Marks:50

Course Objectives:

1. Develop efficient logic using decision-making, loop, and jump control statements.
2. Utilize derived data types like arrays and strings for modular program design.
3. Design and implement modular solutions using functions, recursive logic, pointer operations, and dynamic memory management.
4. Handle complex data structures including structures, unions, and text file operations.

Course Outcomes:

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

1. Apply control flow statements to solve logical and repetitive tasks in C.

2. Implement arrays and string operations to manage and manipulate data efficiently.
3. Design modular code using functions, recursion, and appropriate parameter passing.
4. Utilize pointers and memory operations for effective data handling.
5. Demonstrate competence in dynamic memory allocation and text file processing.

List of Experiments:

1. Write a program to check whether the given number is Armstrong or not.
2. Write a program to find the sum of individual digits of a positive integer.
3. Write a program to generate the first n terms of the Fibonacci sequence.
4. Write a program to find both the largest and smallest number in a list of integer values
5. Write a program to demonstrate change in parameter values while swapping two integers
variables using Call by Value & Call by Address
6. Write a program to perform various string operations.
7. Write a program to search an element in a given list of values.
8. Write a program that uses functions to add two matrices.
9. Write a program to calculate factorial of given integer value using recursive functions
10. Write a program for multiplication of two N X N matrices.
11. Write a program to sort a given list of integers in ascending order.
12. Write a program to calculate the salaries of all employees using Employee (ID, Name, Designation, Basic Pay, DA, HRA, Gross Salary, Deduction, Net Salary) structure.
    a. DA is 30 % of Basic Pay
    b. HRA is 15% of Basic Pay
    c. Deduction is 10% of (Basic Pay + DA)
    d. Gross Salary = Basic Pay + DA+ HRA
    e. Net Salary = Gross Salary - Deduction
13. Write a program to read / write the data from / to a file.
14. Write a program to reverse the contents of a file and store in another file.
15. Write a program to create Book (ISBN,Title, Author, Price, Pages, Publisher) structure and store book details in a file and perform the following operations
    a. Add book details
    b. Search a book details for a given ISBN and display book details, if available
    c. Update a book detail using ISBN
    d. Delete book details for a given ISBN and display list of remaining Books

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

II SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma1-2602 **DATA STRUCTURES USING C** Max.Marks:100

Course Objectives:

1. Understand fundamental concepts of algorithms and data structures with focus on complexity analysis and abstract data types.
2. Explore various types of linked lists and their dynamic memory representations and operations.
3. Analyze and implement linear data structures, such as stacks and queues, and examine their real-world applications.
4. Apply sorting and searching algorithms, understanding their performance implications and optimization strategies.
5. Design and manipulate hierarchical and graph-based structures, applying traversal algorithms and understanding their practical uses in computing.

Course Outcomes: Students will be able to:

1. Explain algorithm characteristics, time and space complexity, and asymptotic notations with clarity.
2. Implement and analyze different types of linked lists, including insertion, deletion, and traversal operations.
3. Develop stack and queue data structures using arrays and linked lists, and apply them in expression evaluation.
4. Apply efficient searching and sorting algorithms to solve computational problems and evaluate performance trade-offs.
5. Construct and traverse tree and graph structures, using them to solve problems like shortest path and spanning trees.

Unit 1. Basic Concepts:

Algorithm: Definition and characteristics, Complexity analysis: Space Complexity, Time Complexity, Asymptotic Notations.

Introduction to Data structures: Definition, Types of Data structures, Abstract Data Types (ADT), Introduction to Linked Lists, Representation of linked lists in Memory, Comparison between Linked List and Array.

Unit 2. Linked Lists:

Types of Linked Lists - Singly Linked list, Doubly Linked list, Circularly Singly Linked list, Circularly Doubly Linked list; Implementation of Single Linked List ADT: Creating a List, Traversing a linked list, Searching in linked list, Insertion and deletion into linked list (At first Node, Specified Position, Last node).

Unit 3. Stacks and Queues:

Introduction to stack ADT, Implementation of stacks using array and Linked List, Application of stacks - Polish Notations - Converting Infix to Post Fix Notation - Evaluation of Post Fix Notation.

Queues: Introduction to Queue ADT, Implementation of Queues using array and Linked List, Application of Queues Types of Queues- Circular Queues, De-queues, Priority Queue, Heaps.

Unit 4. Searching and Sorting:

Linear or Sequential Search, Binary Search, Hashing and collision resolution.

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort

Unit 5. Trees and Graphs:

Tree Terminology, Binary Tree Representation, Traversal techniques, Expression Tree, Binary Search Tree- Definition, Operations on a Binary Search Tree: Creation, Search, Insertion & deletion.

Graphs: Introduction to Graphs, Terminology, Representation (Adjacency Matrix, Adjacency List), Traversal of Graphs (DFS, BFS), Applications of Graphs, Concept of Shortest Path Problems, Concept of Minimum Cost Spanning Tree.

Textbooks:

1. Data Structures Using C, Balagurusamy E. Tata MCGraw Hill
2. Data Structures using C, Reema Thareja, Third Edition, Oxford University Press

Reference Books:

1. Data Structures, Lipschutz, Schaum's Outline Series, Tata Mcgraw-hill
2. Data Structures Using C, Ch. Vijay Kumar, Pen Press International

Activities:

Outcome: Explain algorithm characteristics, time and space complexity, and asymptotic notations with clarity

Activity: Create a comparative chart of algorithms with different notations related to time and space complexities.

Evaluation Method: Rubric-based assessment of the chart for correctness, clarity, and depth of explanation on a 10-point scale.

Outcome: Implement and analyze different types of linked lists, including insertion, deletion, and traversal operations

Activity: Code a menu-driven program in C to implement single linked lists with all basic operations.

Evaluation Method: Practical lab assessment with test cases and Viva-style questioning to explain pointer manipulation.

Outcome: Develop stack and queue data structures using arrays and linked lists, and apply them in expression evaluation

Activity: Build a program to convert infix expressions to postfix and evaluate them using stacks; Implement queues using both arrays and linked lists with enqueue/dequeue operations.

Evaluation Method: Code review and execution of programs for sample cases and evaluation based on correctness and efficiency.

Outcome: Apply efficient searching and sorting algorithms to solve computational problems and evaluate performance trade-offs

Activity: Implement and compare sorting algorithms (e.g., selection sort and bubble sort) and searching algorithms (e.g., Linear vs. Binary Search) on datasets of varying sizes. Record number of swaps and iterations for preparing a chart to assimilate the results.

Evaluation Method: Performance report with graphs and analysis. Oral presentation or peer review discussing trade-offs and algorithm selection rationale.

Outcome: Construct and traverse tree and graph structures, using them to solve problems like shortest path and spanning trees

Activity: Implement binary trees and graphs using adjacency lists/matrices.

Evaluation Method: Lab demo with sample inputs and visual output (e.g., tree traversal order, graph paths).

****

ST. JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

II SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma1-2652 DATA STRUCTURES USING C LAB Max.Marks:50

Course Objectives:

1. Analyze and implement linear data structures, such as stacks and queues.
2. Apply sorting and searching algorithms, understanding their performance implications and optimization strategies.
3. Design and manipulate hierarchical and graph-based structures, applying traversal algorithms and understanding their practical uses in computing.

Course Outcomes: Students will be able to:

1. Implement and analyze different types of linked lists, including insertion, deletion, and traversal operations.
2. Develop stack and queue data structures using arrays and linked lists, and apply them in expression evaluation.
3. Apply efficient searching and sorting algorithms.
4. Construct and traverse tree and graph structures.

List of Experiments

1. Write a program to read 'N' numbers of elements into an array and also perform the

following operation on an array

a. Add an element at the beginning of an array

b. Insert an element at given index of array

c. Update an element using a values and index

d. Delete an existing element

2. Write a program to implement Single Linked List with insertion, deletion and traversal operations

3. Write a program to implement Doubly Linked List with insertion, deletion and traversal operations

4. Write a program to implement the Stack operations using Arrays and Linked Lists.

5. Write a program to convert a given infix expression to a postfix expression using stacks.

6. Write a program to implement the Queue operations using Arrays and Linked Lists.

7. Write a program to implement the Circular Queue operations using Arrays.

8. Write a program for Binary Search Tree Traversals

9. Write a program to search an item in a given list using the following Searching algorithms a. Linear Search b. Binary Search.

10. Write a program for implementation of the following Sorting Algorithms

a. Bubble Sort b. Insertion Sort c. Quick Sort d. Merge Sort

****

ST. JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

II SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma2-2602 **DIGITAL LOGIC DESIGN** Max.Marks:100

Course Objectives

1. Introduce the fundamentals of number systems, their conversions, and binary arithmetic operations.
2. Explore digital logic through gates, Boolean algebra, and simplification techniques for logic functions.
3. Develop proficiency in designing basic combinational circuits like adders and subtractors.
4. Equip students with the skills to implement advanced combinational components such as multiplexers, encoders, and decoders.
5. Foster understanding of sequential circuits, flip-flops, counters, and shift registers for system-level design.

Course Outcomes

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

1. Apply concepts of number systems to perform radix conversions and binary arithmetic using signed and unsigned formats.

2. Simplify logic functions using Boolean algebra, Karnaugh maps, and universal gates.
3. Design and analyze combinational circuits such as half adders, full adders, and subtractors.
4. Construct advanced combinational logic modules, including multiplexers, demultiplexers, encoders, decoders, and their hierarchical versions. Realize complex Boolean functions using combinations of logic modules.
5. Develop and evaluate sequential circuits such as flip-flops, latches, counters, and shift registers.

Unit 1: Number Systems:

Conversion of numbers from one radix to another radix, r's, (r-1)'s complements, signed binary numbers, addition and subtraction of unsigned and signed numbers, weighted and unweighted codes.

Unit 2. Logic Gates and Boolean Algebra:

NOT, AND, OR, universal gates, X-OR and X-NOR gates, Boolean laws and theorems, complement and dual of a logic function, canonical and standard forms, two level realization of logic functions using universal gates, minimizations of logic functions (POS and SOP) using Boolean theorems, K-map (up to four variables), don't care conditions.

Unit 3. Combinational Logic Circuits – 1:

Design of half adder, full adder, half subtractor, full subtractor, ripple adders and subtractors, ripple adder / subtractor.

Unit 4. Combinational Logic Circuits – 2: Design of decoders, encoders, priority encoder, multiplexers, demultiplexers, higher order decoders, demultiplexers and multiplexers, realization of Boolean functions using decoders, multiplexers.

Unit 5. Sequential Logic Circuits:

Classification of sequential circuits, latch and flip-flop, RS- latch using NAND and NOR Gates, RS, JK, T and D flip-flops, truth tables and excitation tables, conversion of flip-flops, flip-flops with asynchronous inputs (preset and clear). Registers- shift registers, bidirectional shift registers, universal shift register, design of ripple counters, modulus counters.

Text Books:

1. Digital Design, M. Morris Mano, Michael D Ciletti,5th edition, Pearson.
2. Digital Logic Design, K.C. Rao, Ramana, Pen International Press

Reference Books:

1. Digital Electronics and Logic Design, Jaydeep Chakravorty, Universities Press
2. Digital Logic Design, Sonali Singh, BPB Publications

Activities:

Outcome: Apply concepts of number systems to perform radix conversions and binary arithmetic using signed and unsigned formats

Activity: Design a calculator in a spreadsheet or simulation tool (e.g., Logisim) that performs: Decimal ↔ Binary ↔ Hexadecimal conversions and binary arithmetic (addition, subtraction).

Evaluation Method: Rubric-based evaluation on a 10point scale (conversion accuracy, arithmetic correctness)

Outcome: Simplify logic functions using Boolean algebra, Karnaugh maps, and universal gates.

Activity: Provide students with complex Boolean expressions and truth tables. Ask them to: Simplify using Boolean laws, Minimize using Karnaugh maps and Implement using only NAND or NOR gates

Evaluation Method: Worksheet submission with step-by-step simplification and evaluation of gate-level implementation using a 10-point scale.

Outcome: Design and analyze combinational circuits such as half adders, full adders, and subtractors

Activity: Build and simulate: Half adder and full adder using logic gates, and half and full subtractor circuits

Evaluation Method: Evaluate the correctness of the circuits for different inputs on a 10-point scale.

Outcome: Construct advanced combinational circuits, including multiplexers, demultiplexers, encoders and decoders.

Activity: Design Multiplexers for function selection, Decoders for control signal generation and Encoders for input compression

Evaluation Method: Project-based evaluation with functional demo and assessments based on a 10-point scale.

Outcome: Develop and evaluate sequential circuits such as flip-flops, latches, counters, and shift registers

Activity: Implement and test SR, JK, D, T flip-flops, asynchronous and synchronous counters using a simulator (E.g. Logisim, Multisim)

Evaluation Method: Lab assessment on a 10-point scale to understand the correctness of the circuit and presentation of the design.

****

ST. JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

II SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma22652 **DIGITAL LOGIC DESIGN LAB** Max.Marks:50

Course Objectives

1. Develop proficiency in designing basic combinational circuits like adders and subtractors.
2. Equip students with the skills to implement combinational components such as multiplexers, encoders, and decoders.

3. Foster understanding of sequential circuits, flip-flops, counters, and shift registers for system-level design.

Course Outcomes

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

1. Design and analyze combinational circuits such as half adders, full adders, and subtractors.
2. Construct advanced combinational logic modules, including multiplexers, demultiplexers, encoders, decoders, and their hierarchical versions.
3. Develop and evaluate sequential circuits such as flip-flops, latches, counters, and shift registers.

List of Experiments

The laboratory work can be done by using physical gates and necessary equipment or simulators.

Simulators: https://sourceforge.net/projects/gatesim/ or https://circuitverse.org/ or any free open- source simulator

1. Introduction to digital electronics lab- nomenclature of digital ICs, specifications, study of the data sheet, concept of Vcc and ground, verification of the truth tables of logic gates using TTL ICs.
2. Implementation of the given Boolean functions using logic gates in both SOP and POS forms .
3. Realization of basic gates using universal gates.
4. Design and implementation of half and full adder circuits using logic gates.
5. Design and implementation of half and full subtractor circuits using logic gates.
6. Verification of stable tables of RS, JK, T and D flip-flops using NAND gates.
7. Implementation and verification of Decoder and encoder using logic gates.
8. Implementation of 4X1 MUX and DeMUX using logic gates.
9. Implementation of 8X1 MUX using suitable lower order MUX.
10. Implementation of 7-segment decoder circuit.
11. Implementation of 4-bit parallel adder.
12. Design and verification of 4-bit modulus counter.

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

III SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma1-3601 Object Oriented Programming using Java Max.Marks:100

Course Objectives:

To introduce the fundamental concepts of Object-Oriented programming and to design & implement object-oriented programming concepts in Java.

Course Outcomes: Students after successful completion of the course will be able to:

1. Demonstrate how object-oriented concepts are incorporated into the Java programming language.[L2]
2. Develop problem-solving and programming skills using Classes & Objects.[L3]
3. Apply the principles of polymorphism , interface and packages concepts.[L3]
4. Develop the ability to solve real-world problems through multithreaded programming using Java.[L3]
5. Develop GUI based applications and web applications.[L3]

UNIT-I

OOPs Concepts and Java Programming: Introduction to Object-Oriented concepts, procedural and object-oriented programming paradigm

Java programming: An Overview of Java, Java Environment, Data types, Variables, constants, scope and life time of variables, operators, type conversion and casting, Accepting Input from the Keyboard, Reading Input with Java.util.Scanner Class, Displaying Output with System.out.printf(), Displaying Formatted Output with String.format(), Control Statements .

UNIT-II

Arrays, Command Line Arguments, Strings-String Class Methods
Classes & Objects: Creating Classes, declaring objects, Methods, parameter passing, static fields and methods, Constructors, and 'this' keyword, overloading methods and access
Inheritance: Inheritance hierarchies, super and subclasses, member access rules, 'super' keyword, preventing inheritance: final classes and methods, the object class and its methods; Polymorphism: Dynamic binding, method overriding, abstract classes and methods;

UNIT-III

Interface: Interfaces VS Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface;

Packages: Defining, creating and accessing a package, understanding CLASSPATH, importing packages.

Exception Handling: Benefits of exception handling, the classification of exceptions, exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes.

UNIT-IV

Multithreading: Differences between multiple processes and multiple threads, thread states, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter thread communication.

Stream based I/O (java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, The Console class, Serialization

UNIT-V

GUI Programming with Swing - Introduction, MVC architecture, components, containers. Understanding Layout Managers - Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout.

Event Handling - The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

Text Books:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

Reference Books

1. Cay S. Horstmann, "Core Java Fundamentals", Volume 1, 11 th Edition, Prentice Hall, 2018.
2. Paul Deitel, Harvey Deitel, "Java SE 8 for programmers", 3rd Edition, Pearson, 2015.
3. S. Malhotra, S. Chudhary, Programming in Java, 2nd edition, Oxford Univ. Press.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz on Object-Oriented Programming Concepts and Java Constructs

Evaluation Method: Quiz Performance and Knowledge Retention

Unit 2: Activity: Object-Oriented Programming Assignment: Class Implementation

Evaluation Method: Assignment Completion and Correctness

Unit 3: Activity: Hands-on Lab Activity: Creating and Using Custom Java Packages

Evaluation Method: Lab Performance and Correctness of Code Implementation

Unit 4: Activity: Case Study Discussion on where multi-threading is crucial

Evaluation Method: Critical thinking, problem-solving, and presentation skills.

Unit 5: Activity: GUI design contest using Java Swings

Evaluation Method: GUI design, Visual appearance and user friendliness, usability, and adherence to event handling principles.

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

III SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma1-3651 Object Oriented Programming using Java Lab Max.Marks:50

Course Objectives:

To introduce the fundamental concepts of Object-Oriented programming and to design & implement object-oriented programming concepts in Java.

Course Outcomes: Students after successful completion of the course will be able to:

1. Demonstrate how object-oriented concepts are incorporated into the Java programming language.[L2]
2. Develop problem-solving and programming skills using Classes & Objects.[L3]
3. Apply the principles of polymorphism , interface and packages concepts.[L3]
4. Develop the ability to solve real-world problems through multithreaded programming using Java.[L3]
5. Develop GUI based applications and web applications.[L3]

List of Experiments

1. Write a Java program to print Fibonacci series using for loop.
2. Write a Java program to calculate multiplication of 2 matrices.
3. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.
4. Write a Java program that implements method overloading.
5. Write a Java program for sorting a given list of names in ascending order.
6. Write a Java program that displays the number of characters, lines and words in a text file.
7. Write a Java program to implement various types of inheritance

i. Single ii. Multi-Level iii. Hierarchical iv. Hybrid

8. Write a java program to implement runtime polymorphism.
9. Write a Java program which accepts withdraw amount from the user and throws an exception "In Sufficient Funds" when withdraw amount more than available amount.
10. Write a Java program to create three threads and that displays "good morning", for every one second, "hello" for every 2 seconds and "welcome" for every 3 seconds by using extending Thread class.
11. Write a Java program that creates three threads. First thread displays "OOPS", the second thread displays "Through" and the third thread Displays "JAVA" by using Runnable interface.
12. Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
13. Implement a Java program for handling key events when the key board is pressed, released, typed.
14. Write a Java swing program that reads two numbers from two separate text fields and display sum of two numbers in third text field when button "add" is pressed.
15. Write a Java program to design student registration form using Swing Controls. The form which having the following fields and button SAVE

Form Fields are: Name, RNO, Mailid, Gender, Branch, Address.

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

III SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma2-3601 **Data Structures using C** Max.Marks:100

Course Objectives: To introduce the fundamental concept of data structures and to emphasize the importance of various data structures in developing and implementing efficient algorithms.

Course Outcomes: Students after successful completion of the course will be able to:

1. Define the concepts related to Data Structures and develop efficient algorithms for solving a problem. [L1].
2. Design and implement linked list concepts.[L6].
3. Discuss basic operations on stacks and queues using array representation. [L6].
4. Design and implement sorting and searching techniques, Summarize the characteristics and applications of different graph traversal techniques [L2,L6].
5. Identify different types of trees and their characteristics.[L3].

UNIT-I

Basic Concepts: Pointers and dynamic memory allocation, Algorithm-Definition and characteristics, Algorithm Analysis-Space Complexity, Time Complexity, Asymptotic Notation.

Introduction to Data structures: Definition, Types of Data structure, Abstract Data Types (ADT), Difference between Abstract Data Types, Data Types, and Data Structures.

Arrays-Concept of Arrays, Single dimensional array, Two dimensional array, Operations on arrays with Algorithms (searching, traversing, inserting, deleting)

UNIT-II

Linked List: Concept of Linked Lists, Representation of linked lists in Memory, Comparison between Linked List and Array, Types of Linked Lists - Singly Linked list, Doubly Linked list, Circularly Singly Linked list, Circularly Doubly Linked list.

Implementation of Linked List ADT: Creating a List, Traversing a linked list, Searching linked list, Insertion and deletion into linked list (At first Node, Specified Position, Last node), Application of linked lists

UNIT-III

Stacks: Introduction to stack ADT, Representation of stacks with array and Linked List, Implementation of stacks, Application of stacks - Polish Notations - Converting Infix to Post Fix Notation - Evaluation of Post Fix Notation - Tower of Hanoi, Recursion: Concept and Comparison between recursion and Iteration

Queues: Introduction to Queue ADT, Representation of Queues with array and Linked List, Implementation of Queues, Application of Queues Types of Queues- Circular Queues, De-queues, Priority Queue

UNIT-IV

Searching: Linear or Sequential Search, Binary Search and Indexed Sequential **Search**

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort

UNIT-V Binary Trees: Concept of Non- Linear Data Structures, Introduction Binary Trees, Types of Trees, Basic Definition of Binary Trees, Properties of Binary Trees,

Representation of Binary Trees, Operations on a Binary Search Tree, Binary Tree Traversal, Applications of Binary Tree.

Graphs: Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs (DFS, BFS), Application of Graphs.

Text Books:

1. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publications Pvt Ltd Delhi India.
2. A.K. Sharma ,Data Structure Using C, Pearson Education India.
3. "Data Structures Using C" Balagurusamy E. TMH

Reference Books

1. "Data Structures through C", Yashavant Kanetkar, BPB Publications
2. Rajesh K. Shukla, "Data Structure Using C and C++" Wiley Dreamtech Publication.
3. Lipschutz, "Data Structures" Schaum's Outline Series, Tata Mcgraw-hill Education (India) Pvt. Ltd .
4. Michael T. Goodrich, Roberto Tamassia, David M. Mount "Data Structures and Algorithms in C++", Wiley India.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Algorithm analysis exercises Evaluation Method: Programming Assignment and Correctness

Unit 2: Activity: Presentations on real-life applications of linked lists

Evaluation Method: Presentation skills or reports

Unit 3: Activity: Role-playing activities for stack operations

Evaluation Method: Problem-solving skills, communication and collaboration abilities.

Unit 4: Activity: Sorting algorithm analysis and comparison activities

Evaluation Method: Performance analysis and presentation.

Unit 5: Activity: Case Study on Applications of Graphs

Evaluation Method: Critical thinking, problem-solving, and presentation skills

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

III SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma2-3661 **Data Structures using C Lab** Max.Marks:50

Course Objectives:

To introduce the fundamental concept of data structures and to emphasize the importance of various data structures in developing and implementing efficient algorithms.

Course Outcomes: Students after successful completion of the course will be able to:

1. Develop programs to perform basic operations (insertion, deletion, traversal) on arrays, linked lists, stacks, and queues in C. [L3,L6].
2. Design and develop algorithms and programs in binary trees and graphs with data structures. [L3,L6].
3. Demonstrate the practical use of sorting, and searching techniques. [L2]

List of Experiments:

1. Write a program to read 'N' numbers of elements into an array and also perform the following operation on an array

   a. Add an element at the beginning of an array

   b. Insert an element at given index of array

   c. Update an element using a values and index

   d. Delete an existing element

2. Write Program to implement Single Linked List with insertion, deletion and traversal operations

3. Write Program to implement Circular doubly Linked List with insertion, deletion and traversal operations

4. Write Programs to implement the Stack operations using an array

5. Write a program using stacks to convert a given infix expression to postfix

6. Write Programs to implement the Stack operations using Liked List.

7. Write Programs to implement the Queue operations using an array.

8. Write Programs to implement the Queue operations using Liked List.

9. Write a program for Binary Search Tree Traversals

10. Write a program to search an item in a given list using the following Searching Algorithms

a. Linear Search b. Binary Search.

11. Write a program for implementation of the following Sorting Algorithms

a. Bubble Sort b. Insertion Sort c. Quick Sort

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

III SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma3-3601 **Computer Organization** Max.Marks:100

Course Objectives:

To familiarize with organizational aspects of memory, processor and I/O..

Course Outcomes: Students after successful completion of the course will be able to:

1. Understand the basics of instructions [L2]
2. Evaluate the performance implications of hierarchical memory organization.[L3]
3. summarize various data transfer techniques.[L2]
4. Demonstrate an understanding of arithmetic operations and illustrate concepts of parallel processing.[L2]
5. Analyze the distinctions between microprogrammed and hard-wired control units.[L4]

UNIT – I

Register Transfer Language and Micro Operations: Introduction- Functional units, computer registers, register transfer language, register transfer, bus and memory transfers, arithmetic, logic and shift micro-operations, arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, instruction cycle. Register reference instructions, Memory – reference instructions, input – output and interrupt.

UNIT – II

CPU and Micro Programmed Control: Central Processing unit: Introduction, instruction formats, addressing modes. Control memory, address sequencing, design of control unit - hard wired control, micro programmed control.

UNIT – III

Memory Organization: Memory hierarchy, main memory, auxiliary memory, associative memory, cache Memory and mappings.

UNIT – IV

Input-Output Organization: Peripheral Devices, input-output interface, asynchronous data transfer, modes of transfer- programmed I/O, priority interrupt, direct memory access, Input – Output Processor (IOP).

UNIT – V

Computer Arithmetic and Parallel Processing: Data representation- fixed point, floating point, addition and subtraction, multiplication and division algorithms. Parallel Processing-Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline.

Text Books:

1. M. Moris Mano, "Computer Systems Architecture", 3rd edition, Pearson/ PHI.

Reference Books:

1. Carl Hamacher, ZvonksVranesic, SafeaZaky, "Computer Organization", 5th edition, McGraw Hill.
2. William Stallings, "Computer Organization and Architecture", 8th edition, Pearson/PHI.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz competition on micro-operations.
Evaluation Method: Accuracy and speed in answering quiz questions.

Unit 2: Activity: Instruction Format Puzzle: Solving a puzzle to decode and understand instruction formats. Evaluation Method: Accuracy and speed in completing the puzzle.

Unit 3: Activity: Memory Hierarchy Poster: Creating informative posters or infographics on memory hierarchy. Evaluation Method: Clarity of information, presentation and creativity of visual design.

Unit 4: Activity: I/O Troubleshooting Challenge Evaluation Method: problem identification, feasibility of proposed solutions, and clarity of explanations.

Unit 5: Activity: Case Study on Parallel processing architecture. Evaluation Method: Understanding of parallel processing concepts and architectures.

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

III SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma3-3651 Computer Organization Lab Max.Marks:50

Course Objectives:

To familiarize with organizational aspects of memory, processor and I/O..

Course Outcomes: Students after successful completion of the course will be able to:

1. Demonstrate proficiency in implementing and analyzing arithmetic micro-operations using logic gates.[L2]
2. Evaluate and compare the effectiveness of different algorithms for binary multiplication, including Booth's algorithm for signed numbers.[L3]
3. demonstrate proficiency in writing assembly language code to compute the expressions using different instruction formats and addressing modes. [L2]

Lab Experiments

1. Implement a C program to convert a Hexadecimal, octal, and binary number to decimal number vice versa.
2. Implement a C program to perform Binary Addition & Subtraction.
3. Implement a C program to perform Multiplication of two binary numbers.

4. Implement arithmetic micro-operations using logic gates.

5. Implement logic and shift micro-operations using logic gates.

6. Implement a C program to perform Multiplication of two binary numbers (signed) using Booth's Algorithms.

7. Implement a C program to perform division of two binary numbers (Unsigned) using restoring division algorithm.

8. Implement a C program to perform division of two binary numbers (Unsigned) using non- restoring division algorithm.

9. Write assembly language code for A+B*(C-D) using various instruction formats in MASM or any open-source assembler.

10. Write assembly language code for A+B*C using various addressing modes in MASM or any open-source assembler.

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

III SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma4-3601 **Operating Systems** Max.Marks:100

Course Objectives:

To gain knowledge about various functions of an operating system like memory management, process management, device management, etc.

Course Outcomes: Students after successful completion of the course will be able to:

1. Demonstrate the structure and design of operating systems. [L2]
2. Compare various algorithms for process scheduling. [L4]
3. Apply various deadlock handling strategies to solve resource allocation problems. [L3]
4. Evaluate the performance of different memory management techniques and page replacement algorithms [L5].
5. Describe file concepts and analyze various disk scheduling strategies. [L4]

UNIT- I

What is Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems– Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

UNIT- II

Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling- Non-Preemptive and Preemptive Scheduling Algorithms.

UNIT III

Process Management: Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

UNIT IV

Memory Management: Physical and Virtual Address Space; Memory Allocation Strategies–Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory.

UNIT V

File and I/O Management, OS security: Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling algorithms.

Text Books:

1. Operating System Principles by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne (7th Edition) Wiley India Edition.

Reference Books

1. Operating Systems: Internals and Design Principles by Stallings (Pearson)
2. Operating Systems by J. Archer Harris (Author), Jyoti Singh (Author) (TMH)

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Case Study on a specific Operating System: highlighting its functions and key features.

Evaluation Method: Case study presentation, depth of understanding of operating system functions, and ability to articulate key concepts.

Unit 2: Activity: Comparison Poster on Scheduling Algorithms

Evaluation Method: Assessment of posters based on content accuracy, clarity of information, visual presentation, and ability to convey key insights.

Unit 3: Activity: Assignment on Dead Lock prevention techniques

Evaluation Method: Understanding, Completion and report.

Unit 4: Activity: Debate on various Memory allocation schemes

Evaluation Method: Debate arguments, ability to counter opposing viewpoints, logical reasoning, and presentation skills.

Unit 5: Activity: Comparative study of various disk scheduling algorithms using real world datasets. Evaluation Method: Analysis methodology, accuracy of results, and presentation of findings and conclusions.

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

III SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma4-3651 **Operating Systems Lab** Max.Marks:50

Course Objectives:

To gain knowledge about various functions of an operating system like memory management, process management, device management, etc.

Course Outcomes: Students after successful completion of the course will be able to:

1. Demonstrate the linux commands.
2. Design and develop process scheduling, deadlock management and file allocation techniques. [L3,L6]
3. Evaluate the performance of different memory management techniques and page replacement algorithms . [L5]

List of Experiments:

1. Illustrate the LINUX commands

a) pwd b) mkdir c) rmdir d) grep e) chmod f) ls g) rm h) cp

2. Write a program to calculate average waiting time and turnaround time of each process using the following CPU Scheduling algorithm for the given process schedules. a) FCFS b) SJF c) Priority d) Round Robin

3. Simulate MVT and MFT memory management techniques

4. Write a program for Bankers Algorithm for Dead Lock Avoidance

5. Implement Bankers Algorithm Dead Lock Prevention.

6. Write a program to simulate Producer-Consumer problem.

7. Simulate all Page replacement algorithms.

a) FIFO b) LRU c) LFU d) Optimal

8. Simulate Paging Techniques of memory management

9. Simulate the following disk scheduling algorithms

a) FCFS b) SSTF c) SCAN d) CSCAN

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

IV SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma1-4601 **Database Management Systems** Max.Marks:100

Course Objectives:

To familiarize with concepts of database design

Course Outcomes: Students after successful completion of the course will be able to:

1. Demonstrate the basics of DBMS and Identify the data models for relevant problems.[L1,L3]
2. Develop entity relationship and convert entity relationship diagrams into RDBMS [L3]
3. Apply the relational database theory to formulate basic SQL queries and relational algebra expressions for the queries.[L3]

4. Describe the basics of SQL and construct queries using SQL.[L3]
5. Apply PL/SQL for processing databases.[L3]

UNIT- I Overview of Database Management System: Introduction to data, information, database,database management systems, file-based system, Drawbacks of file-Based System, database approach, Classification of Database Management Systems, advantages of database approach,Various Data Models, Components of Database Management System, three schema architecture of data base, costs and risks of database approach.

UNIT - II Entity-Relationship Model: Introduction, the building blocks of an entity relationship diagram, classification of entity sets, attribute classification, relationship degree, relationship classification, reducing ER diagram to tables, enhanced entity-relationship model (EERmodel), generalization and specialization, IS A relationship and attribute inheritance,multiple inheritance, constraints on specialization and generalization, advantages of ER modeling.

UNIT - III Relational Model: Introduction, CODD Rules, relational data model, concept of key,relational integrity, relational algebra, relational algebra operations, advantages of relational algebra, limitations of relational algebra, relational calculus, tuple relational calculus, domain relational Calculus (DRC), Functional dependencies and normal forms upto 3rd normal form.

UNIT - IV Structured Query Language: Introduction, Commands in SQL,Data Types in SQL, Data Definition Language, Selection Operation, Projection Operation,Aggregate functions, Data Manipulation Language, Table Modification Commands, JoinOperation, Set Operations, View, Sub Query.

UNIT - V PL/SQL: Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL LanguageElements, Data Types, Operators Precedence, Control Structure, Steps to Create a PL/SQL,Program, Iterative Control, Procedure, Function, Database Triggers, Types of Triggers.

Text Books: 1. Operating System Principles by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne (7th Edition) Wiley India Edition.

Reference Books 1. Database Management Systems by Raghu Ramakrishnan, McGrawhill

2. Principles of Database Systems by J. D. Ullman

3. Fundamentals of Database Systems by R. Elmasri and S. Navathe

4. SQL: The Ultimate Beginners Guide by Steve Tale.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Seminar Presentation on Database Management Systems Evaluation Method:Depth of research, clarity of explanations, ability to address questions and engage the audience.

Unit 2: Activity: Case Study on EER model Evaluation Method:Identification of inheritance relationships, effective use of generalization and specialization, and adherence to constraints.

Unit 3: Activity: Exercise on Normalization: Assign students a set of unnormalized tables and have them normalize the tables to third normal form Evaluation Method:Normalized table designs, identification of functional dependencies, adherence to normalization rules, and elimination of anomalies.

Unit 4: Activity: Competition on SQL Query Writing Evaluation Method:Query correctness, efficiency, proper use of SQL commands, ability to handle complex scenarios, and creativity in query formulation.

Unit 5: Activity: Peer Review of PL/SQL code Evaluation Method:Peer evaluation of code quality, adherence to coding standards, proper use of language elements, and logic.

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

IV SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma1-4651 Database Management Systems Lab Max.Marks:50

Course Objectives:

To familiarize with concepts of database design

Course Outcomes: Students after successful completion of the course will be able to:

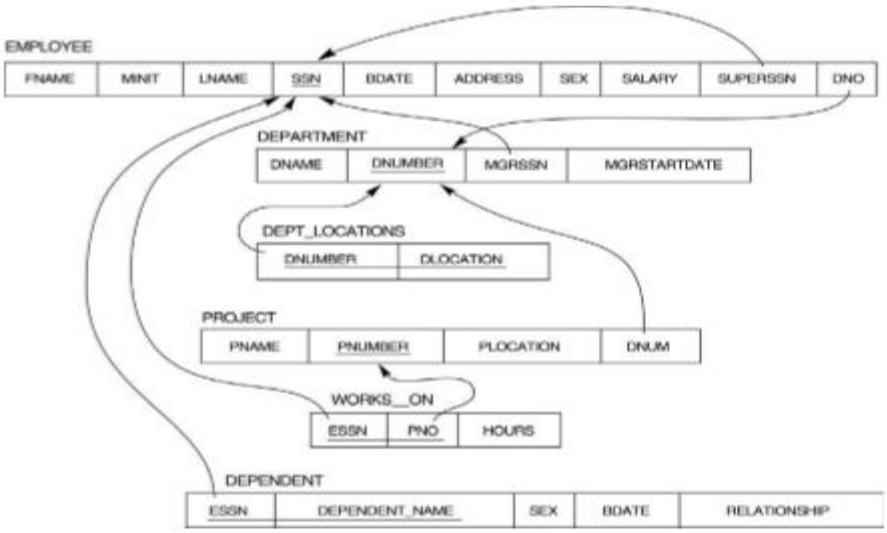
1. Develop entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respective data. [L3]
2. Design queries using SQL. [L6]

List of Experiments:

1. Draw ER diagram for hospital administration

2. Creation of college database and establish relationships between tables

3. Relational database schema of a company is given in the following figure. Relational Database Schema - COMPANY



Questions to be performed on above schema

1. Create above tables with relevant Primary Key, Foreign Key and other constraints 2. Populate the tables with data

3. Display all the details of all employees working in the company.

4. Display ssn,lname,fname, address of employees who work in department no 7.

5. Retrieve the Birthdate and Address of the employee whose name is 'Franklin T. Wong'

6. Retrieve the name and salary of every employee

7. Retrieve all distinct salary values

8. Retrieve all employee names whose address is in _Bellaire'

9. Retrieve all employees who were born during the 1950s

10. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive)

11. Retrieve the names of all employees who do not have supervisors

12. Retrieve SSN and department name for all employees

13. Retrieve the name and address of all employees who work for the 'Research' department

14. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.

15. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.

16. Retrieve all combinations of Employee Name and Department Name

17. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan‘ either as a worker or as a manager of the department that controls the project.

18. Increase the salary of all employees working on the 'ProductX' project by 15%. Retrieve employee name and increased salary of these employees.

19. Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employee first name.

20. Select the names of employees whose salary does not match with salary of any employee in department 10.

21. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford.

22. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with proper headings.

23. Find the sum of the salaries and number of employees of all employees of the _Marketing‘ department, as well as the maximum salary, the minimum salary, and the average salary in this department.

24. Select the names of employees whose salary is greater than the average salary of all employees in department 10.

25. Delete all dependents of employee whose ssn is _123456789‘.

26. Perform a query using alter command to drop/add field and a constraint in Employee table.

ST.JOSEPH’S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

IV SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma2-4601 Object Oriented Software Engineering Max.Marks:100

Course Objectives:

To introduce Object-oriented software engineering (OOSE) - which is a popular technical approach to analyzing, designing an application, system, or business by applying the objectoriented paradigm and visual modeling.

Course Outcomes:

Upon successful completion of the course, a student will be able to:

1. Evaluate and implement the essential ideas of Unified Modeling Language (UML) and Object-Oriented Programming (OOP).[L6]
2. Apply the concepts of object-oriented analysis and design (OOAD), create use cases and scenarios and analyze software requirements.[L3,L6]
3. Make use of test-driven development (TDD) theory and how to use it in practice.[L3]
4. Examine and evaluate strategies for software evolution and maintenance.[L4,L5]
5. Identify Advanced Object-Oriented Software Engineering Concepts.[L3]

UNIT-I Introduction to Object-Oriented Programming: Overview of software engineering, Introduction to Object-Oriented Programming (OOP) concepts (classes, objects, inheritance, polymorphism), Unified Modelling Language (UML) basics, Introduction to software development process and software development life cycle (SDLC).

UNIT-II Requirements Analysis and Design: Requirements analysis and specification, Use cases and scenarios, Object-oriented analysis and design (OOAD), Design patterns, UML modeling techniques (class diagrams, sequence diagrams, state machine diagrams, activity diagrams)

UNIT-III Software Construction and Testing: Software construction basics, Object-oriented design principles, Object-oriented programming languages (Java, C++, Python), Software testing basics (unit testing, integration testing, system testing), Test-driven development (TDD)

UNIT-IV Software Maintenance and Evolution: Software maintenance basics, refactoring techniques Software version control, Code review and inspection, Software evolution and reengineering

UNIT-V Advanced Topics in Object-Oriented Software Engineering: Model-driven engineering (MDE), Aspect-oriented programming (AOP), Component-based software engineering (CBSE), Service Oriented architecture (SOA), Agile software development and Scrum methodologies.

Text Book(s)

1. An Introduction to Object-Oriented Analysis and Design and the Unified Process, 3rd Edition, Craig Larman, Prentice-Hall.
2. Programming in Java by Sachin Malhotra, Oxford University Press

Reference Books

1. Requirements engineering: processes and techniques, G.Kotonya and, I.Sommerville, 1998, Wiley
2. Design Patterns, E.Gamma, R. Helm, R. Johnson, and J. Vlissides
3. The Unified Modeling Language Reference Manual, J. Rumbaugh, I.Jacobson and G. Booch, Addison Wesley

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Group Activity: Design and implement a small OOP project Evaluation Method:Presentation evaluation rubric, Project evaluation based on OOP principles.

Unit 2: Activity: Use Case Scenario Presentation & Peer Activity: Review and provide feedback on each other's use case diagrams Evaluation Method:Presentation evaluation rubric, Peer feedback assessment.

Unit 3: Activity: Poster Presentation: Illustrate TDD principles and benefits Evaluation Method:Poster presentation evaluation

Unit 4: Activity: Peer Activity: Analyze and discuss different maintenance strategies Evaluation Method:Peer discussion participation evaluation

Unit 5: Activity: Seminar on Design Patterns Evaluation Method:Depth of research, clarity of explanations, ability to address questions and engage the audience.

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

IV SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma2-4651 Object Oriented Software Engineering Lab Max.Marks:50

Course Objectives:

To introduce Object-oriented software engineering (OOSE) - which is a popular technical approach to analyzing, designing an application, system, or business by applying the objectoriented paradigm and visual modeling.

Course Outcomes:

Upon successful completion of the course, a student will be able to:

1. Demonstrate proficiency in developing comprehensive System Requirements Specification (SRS) documents.[L2]
2. Analyzing proficiency in designing software architectures using Unified Modeling Language (UML).[L4]

List of Experiments:

Select domain of interest (e.g. College Management System) and identify multi-tier software applications to work on (e.g. Online Fee Collection). Analyze, design and develop this application using OOSE approach:

1. Develop an IEEE standard SRS document. Also develop risk management and project plan (Gantt chart).
2. Understanding of System modeling: Data model i.e. ER – Diagram and draw the ERDiagramwith generalization, specialization and aggregation of specified problem statement
3. Understanding of System modeling: Functional modeling: DFD level 0 i.e. Context Diagram And draw it
4. Understanding of System modeling: Functional modeling: DFD level 1 and DFD level 2 and draw it.
5. Identify use cases and develop the use case model.
6. Identify the business activities and develop an UML Activity diagram.
7. Identity the conceptual classes and develop a domain model with UML Class diagram.
8. Using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams.
9. Draw the state chart diagram.

10. Identify the user interface, domain objects, and technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation. 11. Implement the technical services layer.
12. Implement the domain objects layer.
13. Implement the user interface layer.
14. Draw component and deployment diagrams.

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

IV SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma3-4601 Data Communication and Computer Networks Max.Marks:100

Course Objectives:

To provide students with a comprehensive understanding of networking principles, protocols, and technologies, enabling them to design, analyze, and evaluate efficient and reliable network solutions.

Course Outcomes:

Upon successful completion of the course, a student will be able to:

1. Understand and Contrast the concept of Signals, OSI & TCP/IP reference models.[L2]
2. Discuss and Analyse flow control and error control mechanisms and apply them using standard data link layer protocols.[L4]
3. Design subnets and calculate the IP addresses to network requirements of an organization.[L6]
4. Explain the details of Transport Layer Protocols (UDP, TCP) and suggest appropriate protocol in reliable/unreliable communication.[L2]
5. Analyze the features and operations of various application layer protocols such as HTTP, DNS and SMTP.[L4]

UNIT-I INTRODUCTION: Network applications, network hardware, network software, reference models: OSI, TCP/IP, Internet, Connection oriented network - X.25, frame relay. THE PHYSICAL LAYER: Theoretical basis for communication, guided transmission media, wireless transmission, the public switched telephone networks, mobile telephone system.

UNIT-II THE DATA LINK LAYER: Design issues, error detection and correction, elementary data link protocols, sliding window protocols, example data link protocols - HDLC, the data link layer on the internet. THE MEDIUM ACCESS SUBLAYER: Channel allocations problem, multiple access protocols, Ethernet, Data Link Layer switching, Wireless LAN, Broadband Wireless, Bluetooth.

UNIT-III THE NETWORK LAYER: Network layer design issues, routing algorithms, Congestion control algorithms, Internetworking, the network layer in the internet (IPv4 and IPv6), Quality of Service.

UNIT-IV THE TRANSPORT LAYER: Transport service, elements of transport protocol, Simple TransportProtocol, Internet transport layer protocols: UDP and TCP.

UNIT-V THE APPLICATION LAYER: Domain name system, electronic mail, World Wide Web: architectural overview, dynamic web document and http. APPLICATION LAYER PROTOCOLS: Simple NetworkManagement Protocol, File Transfer Protocol, Simple Mail Transfer Protocol, Telnet.

Text Book(s)

1. S. Tanenbaum (2003), Computer Networks, 4th edition, Pearson Education/ PHI, New Delhi, India Reference Books

2. Behrouz A. Forouzan (2006), Data communication and Networking, 4th Edition, Mc Graw-Hill, India.

3. Kurose, Ross (2010), Computer Networking: A top down approach, Pearson Education, India.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Hands-on exercises to configure network applications Evaluation Method:Practical skills in configuring network applications, hardware, and software.

Unit 2: Activity: Protocol Design and Simulation using simulation tools like NS-3 or Cisco Packet Tracer. Evaluation Method:Students' ability to design and simulate data link layer protocols and multiple access protocols

Unit 3: Activity: Guest Lectures and Workshops on routing algorithms, congestion control, and network layer protocols. Evaluation Method:Students' participation and understanding demonstrated in guest lectures and workshop

Unit 4: Activity: Network Monitoring and Traffic Analysis using tools like Wireshark Evaluation Method:Understanding of transport protocols through their analysis of network traffic and identification of UDP and TCP behavior

Unit 5: Activity: Group Projects on Network Application Development Evaluation Method:Group Project Presentations

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

IV SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma3-4651 Data Communication and Computer Networks Lab Max.Marks:50

Course Objectives:

To provide students with a comprehensive understanding of networking principles, protocols, and technologies, enabling them to design, analyze, and evaluate efficient and reliable network solutions.

Course Outcomes:

Upon successful completion of the course, a student will be able to:

1. Demonstrate understanding of various network tools in Windows and Linux, and apply them to diagnose and troubleshoot network issues effectively.[L2]
2. Analyze and classify different types of network devices and cables.[L4]
3. Design and construct a Local Area Network (LAN) incorporating network devices, cables, and IP addressing schemes.[L6]
4. Evaluate and compare the effectiveness of different routing protocols.[L5]

List of Experiments:

1. Understanding various network tools in Windows and Linux
2. Study different types of Network devices and Cables
3. Building a Local Area Network
4. Concept of Network IP Address
5. Introduction to Network Simulator – Packet Tracer (PT)
6. Configuration of a Router using Packet Tracer
7. Implementation of a Network using Packet Tracer
8. Implementation of Static Routing using Packet Tracer
9. Implementation of RIP using Packet Tracer
10. Implementation of OSPF using Packet Tracer
11. Implement DNS using packet tracer
12. Implementation of a VLAN using Packet Tracer

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

V SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma1-5601 Web Interface Designing Technologies Max.Marks:100

Course Objectives:

To enable students to understand web architecture, develop aesthetic websites, create static and dynamic web pages, implement user interactivity, and gain proficiency in installing and utilizing WordPress and plugins.

Course Outcomes: Students after successful completion of the course will be able to:

1. Understand and appreciate the web architecture and services along with its basic building blocks.[L2]
2. Gain knowledge about various components of a website related to aesthetics.[L2]
3. Make use of the skills regarding creation of a static website and addition of dynamic behavior to a website.[L3]
4. Build experience on making user-interactive web pages.[L6]
5. Learn how to install word press and gain the knowledge of installing various plugins to use in their websites.[L6]

UNIT - I

HTML: Introduction to web designing, difference between web applications and desktop applications, introduction to HTML, HTML structure, elements, attributes, headings, paragraphs, images, tables, lists, blocks, symbols, embedding multi-media components in HTML, HTML forms

UNIT – II

CSS: CSS home, introduction, syntax, CSS combinators, colors, background, borders, margins, padding, height/width, text, fonts, tables, lists, position, overflow, float, pseudo class, pseudo elements, opacity, tool tips, image gallery, CSS forms, CSS counters.

UNIT – III

Java Script: What is DHTML, JavaScript, basics, variables, operators, statements, string manipulations, mathematical functions, arrays, functions. objects, regular expressions, exception handling.

UNIT-IV

Client-Side Scripting: Accessing HTML form elements using Java Script object model, basic data validations, data format validations, generating responsive messages, opening windows using java script, different kinds of dialog boxes, accessing status bar using java script, embedding basic animative features using different keyboard and mouse events.

UNIT – V

Word press: Introduction to word press, features, and advantages, installing and configuring word press and understanding its admin panel (demonstration only), working with posts, managing pages, working with media - Adding, editing, deleting media elements, working with widgets, using menus, working with themes, defining users, roles and profiles, adding external links, extending word press with plug-ins.

Text Book(s)

1. Chris Bates, Web Programming Building Internet Applications, Second Edition, Wiley (2007)

2. Paul S.WangSanda S. Katila, an Introduction to Web Design plus Programming, Thomson (2007).

Reference Books

1. Head First HTML and CSS, Elisabeth Robson, Eric Freeman, O'Reilly Media Inc.

2. An Introduction to HTML and JavaScript: for Scientists and Engineers, David R. Brooks. Springer, 2007

3. Schaum's Easy Outline HTML, David Mercer, Mcgraw Hill Professional.

4. Word press for Beginners, Dr.Andy Williams.

5. Professional word press, Brad Williams, David damstra, Hanstern.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Infographic explaining the necessity to have a web site for each of the agencies such as hotels, hospitals, supermarkets, and educational institutions.

Evaluation Method: Assess the accuracy, visual design, clarity, creativity, use of visual elements, presentation of the infographic explaining the necessity of a website for different agencies.

Unit 2: Activity: Seminar though PPT on various Look and Feel components that websites related to different agencies

Evaluation Method: Content knowledge, organization, clarity, presentation skills, visual aids, audience engagement

Unit 3: Activity: Code snippets Challenge.

Evaluation Method: Accuracy, functionality, efficiency, code readability, and problem-solving approach of the JavaScript code snippets

Unit 4: Activity: Group discussion on different kinds of web forms that take and validate user input using java script validations

Evaluation Method: Active participation, knowledge sharing, critical thinking, and demonstration of different web forms and JavaScript validations

Unit 5: Activity: Creation of Personal website using wordpress

Evaluation Method: Design aesthetics, functionality, user interactivity, content organization, and utilization of plugins.

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

V SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma1-5651 Web Interface Designing Technologies Practical Max.Marks:50

Course Objectives:

To enable students to understand web architecture, develop aesthetic websites, create static and dynamic web pages, implement user interactivity, and gain proficiency in installing and utilizing WordPress and plugins.

Course Outcomes: Students after successful completion of the course will be able to:

1. Apply basic HTML tags to create structured web pages. [L3]
2. Design user-friendly forms and interactive pages using HTML and CSS. [L6]
3. Design user-friendly forms and interactive pages using HTML and CSS. [L6]
4. Develop responsive and visually appealing web pages using styling and layout techniques. [L6]
5. Demonstrate the ability to structure and manage website content using WordPress.[L2]
6. Integrate multimedia and interactive elements to enhance user experience in web pages.[L3]

List of Experiments:

1. Create an HTML document with the following formatting options:
(a) Bold, (b) Italics, (c) Underline, (d) Headings (Using H1 to H6 heading styles), (e) Font (Type, Size and Color), (f) Background (Colored background/Image in background), (g) Paragraph, (h) Line Break, (i) Horizontal Rule, (j) Pre tag
2. Create an HTML document which consists of:
(a) Ordered List (b) Unordered List (c) Nested List (d) Image
3. Create a Table with four rows and five columns. Place an image in one column.
4. Using “table” tag, align the images as follows:

5. Create a menu form using html.
6. Style the menu buttons using CSS.
7. Create a form using HTML which has the following types of controls:
(a) Text Box (b) Option/radio buttons (c) Check boxes (d) Reset and Submit buttons
8. Embed a calendar object in your web page.
9. Create a form that accepts the information from the subscriber of a mailing system.

Word press:

10. Installation and configuration of word press
11. Access admin panel and manage posts
12. Access admin panel and manage pages
13. Add widgets and menus
14. Create users and assign roles
15. Create a site and add a theme to it

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

V SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma2-5601 Max.Marks:100

Web Applications Development using PHP & MYSQL

Course Objectives:

To enable students to understand open-source tools to create dynamic web pages, implement user interactivity, and gain proficiency in developing web sites.

Course Outcomes: Students after successful completion of the course will be able to:

1. Understand and apply the basic constructs of PHP such as variables, data types, operators, flow control statements, and user-defined functions. [L2]
2. Use PHP arrays, string manipulation, date/time functions, and object-oriented concepts to perform dynamic data processing. [L3]
3. Design and develop interactive web forms using PHP for handling input, validating data, file uploads, mail functionality, and exception handling. [L6]
4. Implement session management and cookie handling for maintaining user state and building secure user authentication systems. [L3]
5. Develop dynamic web applications using PHP and MySQL by integrating database connectivity, data manipulation, and CRUD operations. [L6]

UNIT-I

The building blocks of PHP: Variables, Data Types, Operators and Expressions, Constants.

Flow Control Functions in PHP: Switching Flow, Loops, Code Blocks and Browser Output.

Working with Functions: Creating functions, Calling functions, Returning the values from User- Defined Functions, Variable Scope, Saving state between Function calls with the static statement, arguments of functions.

UNIT-II

Working with Arrays: Creating Arrays, Some Array-Related Functions.

Working with Objects: Creating Objects, Accessing Object Instances

Working with Strings, **Dates and Time:** Formatting strings with PHP, Manipulating Strings with PHP, Using Date and Time Functions in PHP.

UNIT-III

Working with Forms: Creating Forms, Accessing Form Input with User defined Arrays, Combining HTML and PHP code on a single Page, Using Hidden Fields to save state, Redirecting the user, Sending Mail on Form Submission, and **Working with File Uploads**, Managing files on server, **Exception handling.**

UNIT-IV

Working with Cookies and User Sessions: Introducing Cookies, setting a Cookie with PHP, Session Function Overview, starting a Session, working with session variables, passing session IDs in the Query String, Destroying Sessions and Unsetting Variables, Using Sessions in an Environment with Registered Users.

UNIT-V

Interacting with MySQL using PHP: MySQL Versus MySQLi Functions, connecting to MySQL with PHP, Working with MySQL Data. Planning and Creating Database Tables,

Creating Menu, Creating Record Addition Mechanism, Viewing Records, Creating the Record Deletion Mechanism.

Text Book(s)

1. Julie C. Meloni, SAMS Teach yourself PHP MySQL and Apache, Pearson Education (2007).
2. Steven Holzner , PHP: The Complete Reference, McGraw-Hill

Reference Books

1. Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition O'reilly, 2014
2. Xue Bai Michael Ekedahl, The web warrior guide to Web Programming, Thomson (2006).

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Infographic explanation of client-server architecture and different server-side scripting languages.
Evaluation Method: Assess the accuracy, visual design, clarity, creativity, use of visual elements, presentation of the infographic explaining the benefits of server-side scripting languages.
Unit 2: Activity: Presentation on various open-source frameworks available in LAMP model
Evaluation Method: Content knowledge, organization, clarity, presentation skills, visual
aids, audience engagement
Unit 3: Activity: Code snippets Challenge.
Evaluation Method: Accuracy, functionality, efficiency, code readability, and problem-solving approach of the PHP code snippets
Unit 4: Activity: Group discussion on Session Management in PHP
Evaluation Method: Active participation, knowledge sharing, critical thinking, and demonstration of Session Management
Unit 5: Activity: Hands-on Lab Session on MYSQL Queries
Evaluation Method: Lab Performance and Correctness of solution Implementation

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

V SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma2-5651 Web Applications Development using PHP & MYSQL Practical

Max.Marks:50

Course Objectives:

To enable students to understand web architecture, develop aesthetic websites, create static and dynamic web pages, implement user interactivity, and gain proficiency in installing and utilizing WordPress and plugins.

Course Outcomes: Students after successful completion of the course will be able to:

1. Understand and apply basic PHP syntax for displaying output, handling variables, and performing date and sequence operations. [L2]
2. Develop PHP scripts to read and process user input and display structured information. [L6]
3. Design and implement interactive forms using HTML and PHP to handle user input and display data on separate PHP pages. [L3]
4. Implement state management techniques such as cookies and sessions using PHP.[L3]
5. Establish database connectivity and perform operations like data retrieval and update using PHP and MySQL. [L6]
6. Collaborate in teams to build a dynamic website using open-source PHP frameworks and MySQL, integrating all learned components. [L6]

List of Experiments:

1. Write a PHP program to Display “Hello”

2. Write a PHP Program to display the today’s date.

3. Write a PHP program to display Fibonacci series.

4. Write a PHP Program to read the employee details.

5. Write a PHP program to prepare the student marks list.

6. Create student registration form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page.

7. Create Website Registration Form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page.

8. Write PHP script to demonstrate passing variables with cookies.

9. Write a PHP script to connect MySQL server from your website.

10. Write a program to keep track of how many times a visitor has loaded the page.

11. Write a PHP application to perform CRUD (Create, Read, Update and Delete) operations on a database table.

12. Create a web site using any open-source framework built on PHP and MySQL – It is a team activity wherein students are divided into multiple groups and each group comes up with their own website with basic features.

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

V SEMESTER	**COMPUTER SCIENCE**	Time: 3Hrs/Week
CS-Ma3-5601	**Foundations of Data Science**	Max.Marks:100

Course Objectives:

To enable students to understand the core concepts and scope of data science, mastering data manipulation and visualization techniques using Python.

Course Outcomes: Students after successful completion of the course will be able to:

1. Identify the need for data science and understand various data collection strategies.[L3]
2. Understand about NoSQL and Descriptive Statistics.[L2]
3. Apply Numpy methods to process the data in an array.[L3]
4. Summarize and Compute Descriptive Statistics using Pandas.[L3]
5. Apply powerful data manipulations visualization using Pandas. [L3]

UNIT-I

Introduction to Data Science: Need for Data Science – What is Data Science - Evolution of Data Science, Data Science Process – Business Intelligence and Data Science – Prerequisites for a Data Scientist – Tools and Skills required. Applications of Data Science in various fields – Data Security Issues.

Data Collection Strategies, Data Pre-Processing Overview, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization, Data Munging, Filtering

UNIT-II

Descriptive Statistics – Mean, Standard Deviation, Skewness and Kurtosis; Box Plots – Pivot Table – Heat Map – Correlation Statistics –ANOVA.

No-SQL: Document Databases, Wide-column Databases and Graphical Databases.

UNIT-III

Python for Data Science –Python Libraries, Python integrated Development Environments (IDE) for Data Science, **NumPy Basics:** Arrays and Vectorized Computation- The NumPy ndarray- Creating ndarrays- Data Types for ndarrays- Arithmetic with NumPy Arrays- Basic Indexing and Slicing - Boolean Indexing- Transposing Arrays and Swapping Axes.

Universal Functions: Fast Element-Wise Array Functions- Mathematical and Statistical Methods-Sorting- Unique and Other Set Logic.

UNIT-IV

Introduction to pandas Data Structures: Series, Data Frame and Essential Functionality: Dropping Entries- Indexing, Selection, and Filtering- Function Application and Mapping- Sorting and Ranking.

Summarizing and Computing Descriptive Statistics- Unique Values, Value Counts, and Membership. Reading and Writing Data in Text Format.

UNIT-V

Data Cleaning and Preparation: Handling Missing Data - Data Transformation: Removing Duplicates, Transforming Data Using a Function or Mapping, Replacing Values, Detecting and Filtering Outliers-

Plotting with pandas: Line Plots, Bar Plots, Histograms and Density Plots, Scatter or Point Plots.

Text Book(s)

1. Y. Daniel Liang, "Introduction to Programming using Python", Pearson, 2012.

2. Wes McKinney, "Python for Data Analysis: Data Wrangling with Pandas, NumPy,and IPython", O'Reilly, 2nd Edition, 2018.

Reference Books

1. Sanjeev Wagh, Manisha Bhende, Anuradha Thakare, 'Fundamentals of Data Science, CRC Press, 1st Edition, 2022

2. Jake VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data", O'Reilly, 2017.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Seminar on Role of Data Science in Politics
Evaluation Method: Content knowledge, organization, clarity, presentation skills, visual aids, audience engagement
Unit 2: Activity: Exercises on Descriptive Statistics
Evaluation Method: Problem Solving, Accuracy
Unit 3: Activity: Hands-on Lab using Numpy
Evaluation Method: Lab Performance and Correctness of solution Implementation
Unit 4: Activity: Hands-on Lab Activity on Pandas
Evaluation Method: Lab Performance and Correctness of solution Implementation.
Unit 5: Activity: Group Activity to visualize college performance records using various plots
Evaluation Method: Active Participation, Post Talk report presentation

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

V SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week

CS-Ma3-5651 Foundations of Data Science Practical Max.Marks:50

Course Objectives:

To enable students to understand the core concepts and scope of data science, mastering data manipulation and visualization techniques using Python.

Course Outcomes: Students after successful completion of the course will be able to:

1. Apply Python data structures and intrinsic NumPy objects to create and manipulate arrays for numerical computing. [L3]
2. Perform numerical and statistical computations on NumPy arrays using universal functions and mathematical methods. [L3]

3. Construct and manipulate Pandas Series and DataFrames using data from various sources including CSV files. [L6]
4. Perform data cleaning, transformation, and conditional operations using Pandas DataFrame functionalities. [L3]
5. Implement vectorized operations and string processing techniques in Pandas for efficient data manipulation. [L3]
6. Visualize datasets using various plotting techniques to gain insights and represent statistical patterns. [L5]

List of Experiments:

1. Study on various python IDEs for Data Science
2. Create NumPy arrays from Python Data Structures, Intrinsic NumPy objects and Random Functions.
3. Manipulation of NumPy arrays- Indexing, Slicing, Reshaping, Joining and Splitting.
4. Computation on NumPy arrays using Universal Functions and Mathematical methods.
5. Create Pandas Series and Data Frame from various inputs.
6. Import any CSV file to Pandas Data Frame and perform the following: a. Visualize the first and last 10 records
b. Get the shape, index and column details
c. Select/Delete the records (rows)/columns based on conditions.
d. Perform ranking and sorting operations.
e. Do required statistical operations on the given column

7. Import any CSV file to Pandas Data Frame and perform the following:
a. Handle missing data by detecting and dropping/ filling missing values.
b. Transform data using apply () and map() method.
c. Detect and filter outliers.
d. Perform Vectorized String operations on Pandas Series.
e. Visualize data using Line Plots, Bar Plots, Histograms, Density Plots and Scatter Plots.

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM

V SEMESTER **COMPUTER SCIENCE** Time: 3Hrs/Week

CS-Ma4-5601 Application Development using Python Max.Marks:100

Course Objectives:

To enable students acquire skills in Python syntax, data structures, object-oriented programming, to interact with databases, handle files, and build graphical user interfaces.

Course Outcomes: Students after successful completion of the course will be able to:
1. Examine Python syntax and semantics and be fluent in the use of Python flow control and functions. [L4]
2. Demonstrate proficiency in handling Strings and File Systems.[L2]
3. Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions. [L6]

4. Interpret the concepts of Web Programming and GUI in Python. [L5]
5. Apply concepts of Python programming in various fields related to IOT, Web Services and Databases in Python.[L3]

UNIT-I

Python basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types

Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules

Sequences - Strings, Lists, and Tuples, Dictionaries and Set Types

Control Flow, Truthiness, Sorting, List Comprehensions, Generators and Iterators

UNIT-II

Files: File Objects, File Built-in Function [open()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution

Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management, Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, Creating Exceptions, Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module, Related Modules

Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages, Other Features of Modules

UNIT-III

Regular Expressions: Introduction, Special Symbols and Characters, Res and Python

Multithreaded Programming: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules

UNIT-IV

GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs

Web Programming: Introduction, Wed Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application, Advanced CGI, Web (HTTP) Servers

UNIT-V

Database Programming: Introduction, Python Database Application Programmer's Interface (DBAPI), Object Relational Managers (ORMs), Related Modules

Text Book(s)

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.
2. Think Python, Allen Downey, Green Tea Press.

Reference Books

1. Introduction to Python, Kenneth A. Lambert, Cengage.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson.
3. Learning Python, Mark Lutz, O' Really.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Hands-on Lab exercise on Python Control Statements

Evaluation Method: Lab Performance and Correctness of solution Implementation
Unit 2: Activity: Assignment of Files in Python
Evaluation Method: Problem Solving, Accuracy
Unit 3: Activity: Exercises on Regular expressions
Evaluation Method: Solutions, Accuracy of Validation
Unit 4: Activity: Poster Presentation on various GUI components in Python
Evaluation Method: Content knowledge, organization, clarity, presentation skills, visual aids.
Unit 5: Activity: Group Project
Evaluation Method: Project effectiveness, User interface, Solution to the Problem

****

ST.JOSEPH'S COLLEGE FOR WOMEN (AUTONOMOUS), VISAKHAPATNAM
V SEMESTER **COMPUTER SCIENCE** Time: 2Hrs/Week
CS-Ma4-5651 Application Development using Python Practical
Max.Marks:50

Course Objectives:
To enable students acquire skills in Python syntax, data structures, object-oriented programming, interacting with databases, handling files, and build graphical user interfaces.

Course Outcomes: Students after successful completion of the course will be able to:
1. Demonstrate problem-solving skills using Python programming constructs such as decision-making, loops, and functions.[L2]
2. Understand and apply sequence data types (lists, tuples, strings, and dictionaries) for data manipulation.[L2]
3. Apply concepts of object-oriented and concurrent programming using Python.[L3]
4. Implement basic GUI applications using Tkinter library in Python.[L3]
5. Perform file handling operations to read and manipulate data.[L3]
6. Demonstrate the use of Python in database programming for performing CRUD operations.[L2]

List of Experiments:
1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice.

2. Write a python program to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria :
Grade A: Percentage >=80 Grade B: Percentage>=70 and 80
Grade C: Percentage>=60 and <70 Grade D: Percentage>=40 and <60 Grade E: Percentage<40
3. Demonstrate various methods of Sequence Data Types

4. Write a python program to display the first n terms of Fibonacci series.

5. Write a python program to calculate the sum and product of two compatible matrices.

6. Write a function that takes a character and returns True if it is a vowel and False otherwise.

7. Write a program to implement exception handling.

8. Write a program to implement Multithreading

9. Develop a Python GUI calculator using Tkinter

10. Write a Python program to read last 5 lines of a file.

11. Design a simple database application that stores the records and retrieve the same

12. Design a database application to search the specified record from the database.

13. Design a database application to that allows the user to add, delete and modify the records.

****