

Computer Science Syllabus Grid 2024-25

SEMESTER	SUBJECT
II	Problem Solving using C(T)(Major) Problem Solving using C (P) Digital Logic Design(T) Digital Logic Design(P) Problem Solving using C(T)(Minor) Problem Solving using C (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	Data Science(T) Data Science(P) Python for Data Science(T) Python for Data Science(P)
VI	Semester Internship

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

II SEMESTER

COMPUTER SCIENCE

Time: 3Hrs/Week

CS-Ma1-2601

Problem Solving using C

Max.Marks:100

Course Objectives

1. To explore basic knowledge on computers
2. Learn how to solve common types of computing problems.
3. Learn to map problems to programming features of C.
4. Learn to write good portable C programs.

Course Outcomes

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

1. Understand the working of a digital computer and Fundamental constructs of Programming
2. Analyze and develop a solution to a given problem with suitable control structures
3. Apply the derived data types in program solutions
4. Use the 'C' language constructs in the right way
5. Apply the Dynamic Memory Management for effective memory utilization

UNIT-I

Introduction to computer and programming: Introduction, Basic block diagram and functions of various components of computer, Concepts of Hardware and software, 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-II

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 go to.

UNIT-III

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-IV

Functions: Function Prototype, definition and calling. Return statement. Nesting of functions. Categories of functions. Recursion, Parameter Passing by address & by value. Local and Global variables. Storage classes: automatic, external, static and register.

Pointers: Pointer data type, Pointer declaration, initialization, accessing values using pointers. Pointer arithmetic. Pointers and arrays, pointers and functions.

UNIT-V

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.

Text Books:

1. E. Balagurusamy, "Programming in ANSI C", Tata McGraw Hill, 6th Edn, ISBN-13: 978-1-25- 90046-2
2. Herbert Schildt, –Complete Reference with C, Tata McGraw Hill, 4th Edn., ISBN- 13:9780070411838, 2000
3. Computer fundamentals and programming in C, REEMA THAREJA, OXFORD UNIVERSITY PRESS

Reference Books

1. E Balagurusamy, COMPUTING FUNDAMENTALS & C PROGRAMMING – Tata McGraw-Hill, Second Reprint 2008, ISBN 978-0-07-066909-3.
2. Ashok N Kamthane, Programming with ANSI and Turbo C, Pearson Edition Publ, 2002.
3. Henry Mullish&Huubert L.Cooper: The Spirit of C An Introduction to modern Programming, Jaico Pub. House,1996.
4. Y kanithkar, let us C BPB, 13th edition-2013, ISBN:978-8183331630,656 pages

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

II SEMESTER

COMPUTER SCIENCE

Time: 3Hrs/Week

CS-Ma2-2601

Digital Logic Design

Max.Marks:100

Course Objectives:

To familiarize with the concepts of designing digital circuits.

Course Outcomes

Upon successful completion of the course, the students will be able to

1. Understand how to Convert numbers from one radix to another radix and perform arithmetic operations.
2. Simplify Boolean functions using Boolean algebra and k- maps
3. Design adders and subtractors circuits
4. Design combinational logic circuits such as decoders, encoders, multiplexers and demultiplexers.
5. Use flip flops to design registers and counters.

UNIT – I

Number Systems: Binary, octal, decimal, hexadecimal 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 – II

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 – III

Combinational Logic Circuits – 1: Design of half adder, full adder, half subtractor, full subtractor, ripple adders and subtractors, ripple adder / subtractor.

UNIT – IV

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 – V

Sequential Logic Circuits: Classification of sequential circuits, latch and flip-flop, RS- latch using NAND and NOR Gates, truth tables, RS, JK, T and D flip-

flops, truth and excitation tables, conversion of flip- flops, flip-flops with asynchronous inputs (preset and clear).

Design of registers, shift registers, bidirectional shift registers, universal shift register, design of ripple counters, synchronous counters and variable modulus counters.

Text Books:

1. M. Morris Mano, Michael D Ciletti, “Digital Design”, 5th edition, PEA.

Reference Books

1. Kohavi, Jha, “Switching and Finite Automata Theory”, 3rd edition, Cambridge.
2. Leach, Malvino, Saha, “Digital Principles and Applications”, 7th edition, TMH.
3. Roth, “Fundamentals of Logic Design”, 5th edition, Cengage

Course Objectives

1. To explore basic knowledge on computers
2. Learn how to solve common types of computing problems.
3. Learn to map problems to programming features of C.
4. Learn to write good portable C programs.

Course Outcomes

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

1. Understand the working of a digital computer and Fundamental constructs of Programming
2. Analyze and develop a solution to a given problem with suitable control structures
3. Apply the derived data types in program solutions
4. Use the 'C' language constructs in the right way
5. Apply the Dynamic Memory Management for effective memory utilization

UNIT-I

Introduction to computer and programming: Introduction, Basic block diagram and functions of various components of computer, Concepts of Hardware and software, 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-II

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 go to.

UNIT-III

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-IV

Functions: Function Prototype, definition and calling. Return statement. Nesting of functions. Categories of functions. Recursion, Parameter Passing by address & by value. Local and Global variables. Storage classes: automatic, external, static and register.

Pointers: Pointer data type, Pointer declaration, initialization, accessing values using pointers. Pointer arithmetic. Pointers and arrays, pointers and functions.

UNIT-V

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.

Text Books:

1. E. Balagurusamy, "Programming in ANSI C", Tata McGraw Hill, 6th Edn, ISBN-13: 978-1-25- 90046-2
2. Herbert Schildt, –Complete Reference with C, Tata McGraw Hill, 4th Edn., ISBN- 13:9780070411838, 2000
3. Computer fundamentals and programming in C, REEMA THAREJA, OXFORD UNIVERSITY PRESS

Reference Books

1. E Balagurusamy, COMPUTING FUNDAMENTALS & C PROGRAMMING – Tata McGraw-Hill, Second Reprint 2008, ISBN 978-0-07-066909-3.
2. Ashok N Kamthane, Programming with ANSI and Turbo C, Pearson Edition Publ, 2002.
3. Henry Mullish&Huubert L.Cooper: The Spirit of C An Introduction to modern Programming, Jaico Pub. House,1996.
4. Y kanithkar, let us C BPB, 13 th edition-2013, ISBN:978-8183331630,656 pages

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.

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.

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-Ma2-3651

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

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.

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.

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.

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

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:

6. Demonstrate how object-oriented concepts are incorporated into the Java programming language. [L2]
7. Develop problem-solving and programming skills using Classes & Objects. [L3]
8. Apply the principles of polymorphism, interface and packages concepts. [L3]
9. Develop the ability to solve real-world problems through multithreaded programming using Java. [L3]
10. 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.

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:

6. Demonstrate how object-oriented concepts are incorporated into the Java programming language.[L2]
7. Develop problem-solving and programming skills using Classes & Objects.[L3]
8. Apply the principles of polymorphism , interface and packages concepts.[L3]
9. Develop the ability to solve real-world problems through multithreaded programming using Java.[L3]
10. 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

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, Join Operation, Set Operations, View, Sub Query.

UNIT - V PL/SQL: Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL Language Elements, 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:

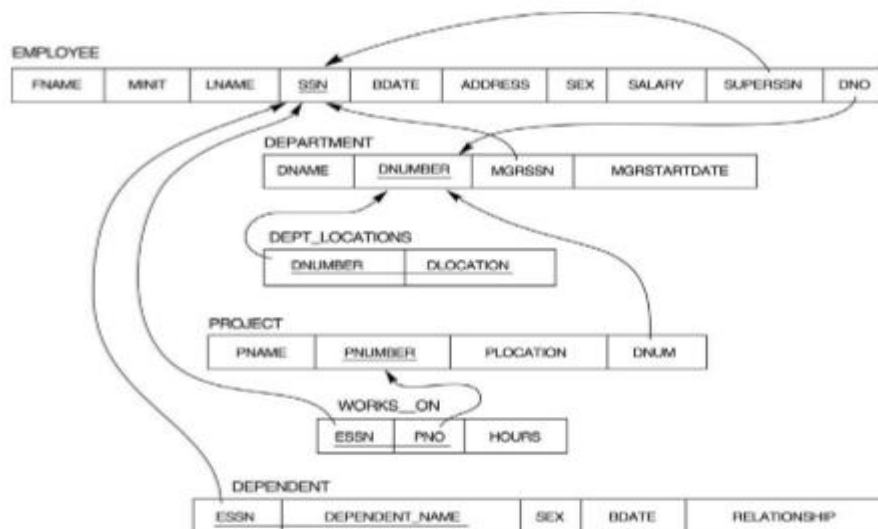
To familiarize with concepts of database design

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 object oriented 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 Transport Protocol, 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 Network Management 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

Course Objectives:

To familiarize with concepts of database design

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

6. Demonstrate the basics of DBMS and Identify the data models for relevant problems.[L1,L3]
7. Develop entity relationship and convert entity relationship diagrams into RDBMS [L3]
8. Apply the relational database theory to formulate basic SQL queries and relational algebra expressions for the queries.[L3]
9. Describe the basics of SQL and construct queries using SQL.[L3]
10. 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, Join Operation, Set Operations, View, Sub Query.

UNIT - V PL/SQL: Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL Language Elements, 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.

Course Objectives:

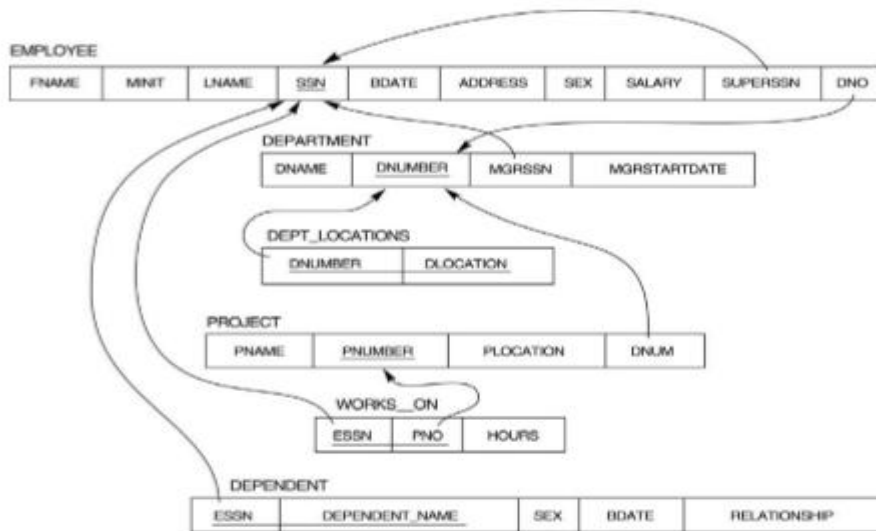
To familiarize with concepts of database design

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

3. Develop entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respective data. [L3]
4. 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.

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 object oriented 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-Mi2-4651 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 object-oriented 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 ER Diagram with 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. Identify 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
V SEMESTER **COMPUTER SCIENCE** Time:3Hrs/Week
CS-E3-5603 **DATA SCIENCE** Max.Marks:100
w.e.f. ("20AH") **SYLLABUS**

Course Objectives: -

- To learn the basics of data science
- Implement it in real-world environment
- Learn basic programming using data science.

Course Outcomes:

- Students after successful completion of the course will be able to:
- Develop programming abilities.
- Demonstrate statistical analysis of data.
- Develop the ability to build data-based models.
- Improve their skill in data management.
- Apply data science concepts and methods to solve problems in real-world contexts.

UNIT I

Introduction: The Ascendance of Data, What is Data Science? , Finding key Connectors, Data Scientists You May Know, Salaries and Experience, Paid Accounts, Topics of Interest, Onward.

Python: Getting Python, The Zen of Python, Whitespace Formatting, Modules, Arithmetic, Functions, Strings, Exceptions, Lists, Tuples, Dictionaries, Sets, Control Flow, Truthiness, Sorting, List Comprehensions, Generators and Iterators, Randomness, Object – Orienting Programming, Functional Tools, enumerate, zip and Argument Unpacking, args and kwargs, Welcome to Data Sciencester!

Visualizing Data: matplotlib, Bar charts, Line charts, Scatterplots.

Linear Algebra: Vectors, Matrices

UNIT II

Statistics: Describing a Single Set of Data, Correlation, Simpson's Paradox, some Other Correlation Caveats, Correlation and Causation.

Probability: Dependence and Independence, Conditional Probability, Bayes's Theorem, Random Variables, Continuous Distributions, The Normal Distribution, The Central Limit Theorem.

Hypothesis and Inference: Statistical Hypothesis Testing, Example: Flipping a Coin, Confidence Intervals, P-hacking, Example: Running an A/B Test, Bayesian Inference.

Gradient Descent: The Idea behind Gradient Descent, Estimating the Gradient, Using the Gradient, Choosing the Right Step Size, Putting It All Together, Stochastic Gradient Descent.

UNIT III

Getting Data: stdin and stdout, Reading Files – The Basics of Text Files, Delimited Files, Scraping the Web - HTML and the parsing Thereof, Example: O’Reilly Books About Data, Using APIs – JSON (and XML), Using an Unauthenticated API, Finding APIs.

Working with Data: Exploring Your Data, Exploring One-Dimensional Data, Two Dimensions Many Dimensions, Cleaning and Munging, Manipulating Data, Rescaling, Dimensionality Reduction.

UNIT IV

K-Nearest Neighbors: The Model, Example: Favorite Languages, The Curse of Dimensionality.

Naive Bayes: A Really Dumb Spam Filter, A More Sophisticated Spam Filter, Implementation, Testing Our Model.

Simple Linear Regression: The Model, Using Gradient Descent, Maximum Likelihood Estimation.

Multiple Regression: The Model, Further Assumptions of the Least Squares Model, Fitting the Model, Interpreting the Model, Goodness of Fit.

UNIT V

Logistic Regression: The Problem, The Logistic Function, Applying the Model, Goodness of Fit Support Vector Machines.

Decision Trees: What Is a Decision Tree? Entropy, The Entropy of a Partition, Creating a Decision Tree, Putting It All Together, Random Forests.

Neural Networks: Perceptron, Feed-Forward Neural Networks And Back propagation, Example: Defeating a CAPTCHA.

Clustering: The Idea, The Model, Example: Meetups , Choosing k, Example: Clustering Colors, Bottom-up Hierarchical Clustering.

References

1. Data Science from Scratch by Joel Grus O’Reilly Media
2. Wes McKinney, “Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython”, O’Reilly, 2nd Edition, 2018.
3. Jake VanderPlas, “Python Data Science Handbook: Essential Tools for Working with Data”, O’Reilly, 2017.
4. Web resources:
 - a. <https://www.edx.org/course/analyzing-data-with-python>
 - b. [http://math.ecnu.edu.cn/~lfzhou/seminar/\[Joel_Grus\]_Data_Science_from_Scratch_First_Princ.pdf](http://math.ecnu.edu.cn/~lfzhou/seminar/[Joel_Grus]_Data_Science_from_Scratch_First_Princ.pdf)
5. Other web sources suggested by the teacher concerned and the college librarian including reading material.

Course objectives:

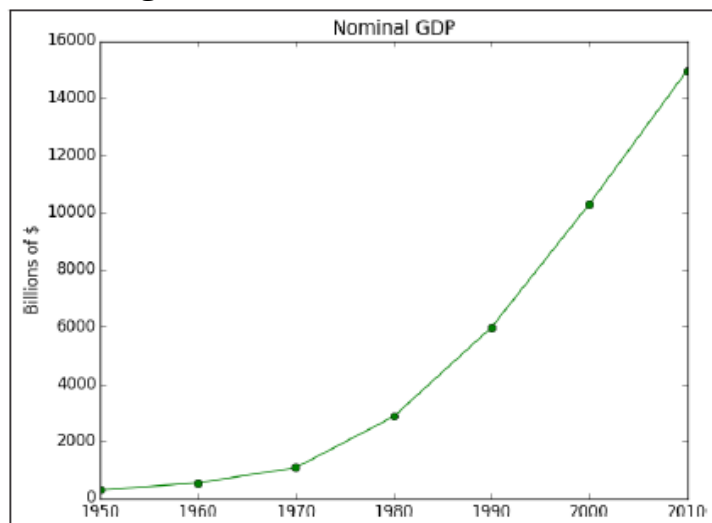
- Understand the basics of Data Science
- To learn how to solve real-world problems using python
- Gain knowledge about the installation of python software

Course Outcomes: On successful completion of this practical course, student shall be able to:

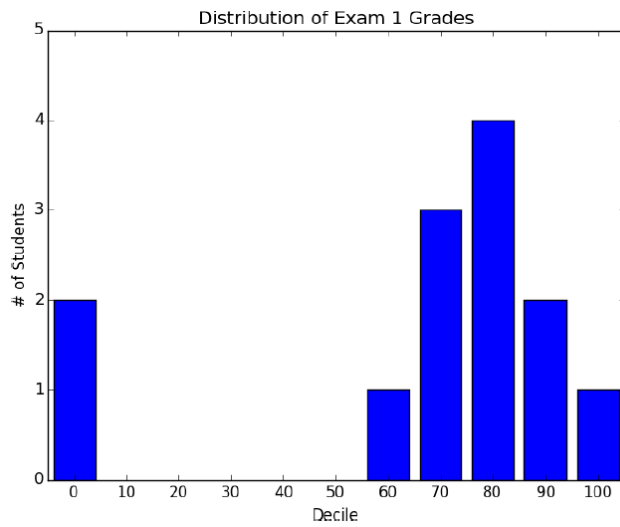
- Apply data science solutions to real world problems.
- Implement the programs to get the required data and present the outputs using Python language.
- Execute statistical analysis with open python software.

Practical (Laboratory) Syllabus

1. Write a Python program to create a line chart for values of year and GDP as given below



2. Write a Python program to create a bar chart to display number of students secured different grading as given below



3. Write a Python program to create a time series chart by taking one year month wise stock data in a CSV file
4. Write a Python program to plot distribution curve
5. Import a CSV file and perform various Statistical and Comparison operations on rows/columns. Write a python program to plot a graph of people with pulse rate p vs. height h . The values of P and H are to be entered by the user.
6. Import rainfall data of some location with the help of packages available in R Studio and plot a chart of your choice.

Course Objectives: -

- To learn various python concepts, functions and packages
- To enable data manipulation and analysis.
- To learn how to solve real-world problems using python.

Course Outcomes:

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

- Learning the need for data science using python.
- Solve basic problems using Python built-in data types and their methods.
- To demonstrate an application with user-defined modules and packages using OOP concept
- Implement data operations using NumPy arrays for easy storage.
- Applying data manipulation methods using Pandas.
- Implement data pre-processing and visualization of applications using Pandas

UNIT - I

Introduction to Data Science - Why Python? - Essential Python libraries - Python Introduction- Features, Identifiers, Reserved words, Indentation, Comments, Built-in Data types and their Methods: Strings, List, Tuples, Dictionary, Set - Type Conversion- Operators.

Decision Making- Looping- Loop Control statement- Math and Random number functions. User defined functions - function arguments & its types.

UNIT -II

User defined Modules and Packages in Python- Files: File manipulations, File and Directory related methods - Python Exception Handling.

OOPs Concepts -Class and Objects, Constructors – Data hiding- Data Abstraction- Inheritance.

UNIT -III

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- String Manipulation: Vectorized String Functions in pandas. Plotting with pandas: Line Plots, Bar Plots, Histograms and Density Plots, Scatter or Point Plots.

References : -

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.
3. Jake VanderPlas, “Python Data Science Handbook: Essential Tools for Working with Data”, O’Reilly, 2017.
4. Wesley J. Chun, “Core Python Programming”, Prentice Hall, 2006.
5. Mark Lutz, “Learning Python”, O’Reilly, 4th Edition, 2009.
6. Web resources:
 - a. <https://www.edx.org/course/python-basics-for-data-science>
 - b. <https://www.edx.org/course/analyzing-data-with-python>
 - c. <https://www.coursera.org/learn/python-plotting?specialization=data-science-python>
 - d. <https://www.programmer-books.com/introducing-data-science-pdf/>
 - e. <https://www.cs.uky.edu/~keen/115/Haltermanpythonbook.pdf>
7. Other web sources suggested by the teacher concerned and the college librarian including reading material.

Course objectives:

- Understand the basics and syntax of Python programming
- To learn how to solve real-world problems using python
- Gain knowledge about the installation of python software

Course Outcomes:

On successful completion of this practical course, student shall be able to:

- Implement simple programs in Python.
- Demonstrate programs related to various structures like arrays, lists, Data frames, etc.
- Implement programs related and using files.
- Implement applications related to data science.

Practical (Laboratory) Syllabus

1. Perform Creation, indexing, slicing, concatenation and repetition operations on Python built-in data types: Strings, List, Tuples, Dictionary, Set
2. Apply Python built-in data types: Strings, List, Tuples, Dictionary, Set and their methods to solve any given problem.
3. Handle numerical operations using math and random number functions
4. Create user-defined functions with different types of function arguments.
5. Create packages and import modules from packages.
6. Perform File manipulations- open, close, read, write, append and copy from one file to another.
7. Write a program for Handle Exceptions using Python Built-in Exceptions
8. Write a program to implement OOP concepts like Data hiding and Data Abstraction.
9. Create NumPy arrays from Python Data Structures, Intrinsic NumPy objects and Random Functions.
10. Manipulation of NumPy arrays- Indexing, Slicing, Reshaping, Joining and Splitting.
11. Computation on NumPy arrays using Universal Functions and Mathematical methods.
12. Load an image file and do crop and flip operation using NumPy Indexing.
13. Create Pandas Series and Data Frame from various inputs.
14. 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 columns.

- (f) Find the count and uniqueness of the given categorical values.
- (g) Rename single/multiple columns

15. 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.