Syllabi of
Departmental Courses (DC) Courses
B.Tech II Semester
For batch admitted 2022-23
(Computer Science and Engineering)
Under Flexible Curriculum

Department of Computer Science and Engineering

DATA STRUCTURES
(150221) (DC)

COURSE OBJECTIVES

- To be familiar with the use of data structures as the foundational base for computer solutions to problems.
- To understand various techniques of searching and sorting.
- To understand basic concepts about stacks, queues, lists, trees and graphs.

Unit-I

Introduction to Data Structures: Algorithms & their characteristics, asymptotic notations. arrays and its representations, index to address translation. **Link list:** Introduction, implementation of linked list, operations, circular link list, doubly linked list, polynomial manipulation using linked list.

Unit-II

Stacks: Concepts and implementation of stacks, operations on stack, conversion of infix to postfix notation, evaluation of postfix expression, recursion.

Queues: Concepts and implementation, operations on queues, dequeue, priority queues, circular queues and application.

Unit-III

Trees: Types, terminology, binary tree -representations, traversal, conversion of general tree to binary tree, binary search tree, threaded binary tree and height balanced tree.

Unit-IV

Searching & Sorting: Linear search, binary search, bubble sort, selection sort, insertion sort, quick sort, merge sort, radix sort and heap sort, comparison between sorting techniques, hashing and collision resolution techniques.

Unit-V

Graphs: Background, graph theory terminologies, representation of graphs- sequential & linked representation, path matrix, graph traversals- BFS, DFS, spanning trees, applications of graph.

RECOMMENDED BOOKS

- Data Structures, Algorithms and Applications in C++, Sartaj Sahni, 2nd Edition.
- An Introduction to Data Structures with Applications, Jean-Paul Tremblay, Mcgraw hill.
- Data Structures & Algorithms, Aho, Hopcroft & Ullman, original edition, Pearson Publication.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1. outline the basics of Algorithms and their performance criteria's.

CO2. explain the working of linear/Non Linear data structures.

CO3. identify the appropriate data structure to solve specific problems.

CO4. analyze the performance of various Data Structures & their applications.

CO5. evaluate the time/space complexities of various data structures & their applications.

CO6. design the optimal algorithmic solutions for various problems.

Data Structure
Experiment List

Write C/C++ Programs to illustrate the concept of the following:

1. Implementation of Array and linked list.
2. Implementation of Sorting Algorithms-Non-Recursive and Recursive.
3. Implementation of Searching Algorithms-Linear and Binary Search.
4. Implementation of Stack using Array.
5. Implementation of Queue using Array.
6. Implementation of Circular Queue using Array.
7. Implementation of Stack using Linked List.
8. Implementation of Queue using Linked List.
9. Implementation of Circular Queue using Linked List.
10. Implementation of Tree Structures, Binary Tree, Tree Traversal, Binary Search Tree, Insertion and Deletion in BST.
11. Graph Implementation, BFS, DFS, Minimum cost spanning tree, shortest path algorithm.

Data Structure
skill-based project list

1. Develop an application to calculate the address of given index position of 2D,3D,4D and nD array elements.
2. Develop an utility to store polynomial equations and to add two polynomial equations.
3. Design the solution of Tower of Hanoi mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

a) Only one disk can be moved at a time.
b) Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
c) No disk may be placed on top of a smaller disk. Develop an application to generate preoder, postorder and inorder traversing sequences of given binary tree.

4. Design a Palindrome-Checker algorithm and implement the utility for Palindrome checking. A palindrome is a string that reads the same forward and backward, for example, radar, toot, and madam.
5. Design a solution of the Word Ladder Problem and develop a software. Example - Transform the word "FOOL" into the word "SAGE". In a word ladder puzzle, you must make the change occur gradually by changing one letter at a time. At each step you must transform one word into another word, you are not allowed to transform a word into a non-word.
6. Develop an application to solve Single-Source Shortest Path (SSSP) problem All-Pairs Shortest Path (APSP).
7. Find out Longest Increasing Subsequence in the given elements. The Longest Increasing Subsequence (LIS) problem is to find the length of the longest subsequence of a given sequence such that all elements of the subsequence are sorted in increasing order. For example, the length of LIS for {10, 22, 9, 33, 21, 50, 41, 60, 80} is 6 and LIS is {10, 22, 33, 50, 60, 80}.

Please Note: *Each project has to be submitted by a group of 3 to 4 students, and each group will be assigned only one project.*

* * * * * * * * * * *

Department of Computer Science and Engineering

Python Programming (150222)

COURSE OBJECTIVES

- To understand the structure and components of a Python program.
- To learn the basic construct of python programming for implementing interdisciplinary research-based problems.
- To plot data using appropriate Python visualization libraries for analysis.

Unit I

Introduction to Python: Setting up programming environment, running python programs from a terminal, variables and simple data types: variables, strings, numbers and maths, comments, conditional statements, introducing loops, working of input function.

Unit II

Tuples and Lists: Tuples, lists, list operations, using if statements with lists, organizing a list, working with lists: looping through an entire list, making numeric lists, working with part of a list. dictionaries and sets: simple dictionary, looping through a dictionary, nesting, example with a dictionary, fibonacci and dictionaries, global variables, defining a set, set operations.

Unit III

Functions: Defining a function, passing arguments, return values, passing a list, passing an arbitrary number of arguments, storing your functions in module, in built functions, lambda functions. classes and inheritance: object oriented programming, creating and using a class, working with class instances, methods, inheritance, importing classes, python standard library.

Unit IV

Files and Exceptions: Reading from a file, writing to a file, file operations, assertions, exceptions, exception example. debugging: programming challenges, classes of tests, bugs, and debugging, debugging examples.

Unit V

Data Visualization: Installing matplotlib, plotting a simple line graph, random walks, making histogram. graphical user interfaces: event-driven programming paradigm; tkinter module, creating simple gui; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors, layouts, nested frames.

RECOMMENDED BOOKS

- Python Crash Course: A Hands-On, Project-Based Introduction to Programming, By Eric Matthes
- Learn Python the Hard Way: 3rd Edition
- T.R. Padmanabhan, Programming with Python, Springer, 1st Ed., 2016.
- Kenneth Lambert, Fundamentals of Python: First Programs, Cengage Learning, 1st Ed., 2012.

COURSE OUTCOMES

After completion of this course, the students will be able to:

CO1. understand basic python programming constructs **CO2.**
analyze various data structures available in python
CO3. implement the Object-oriented programming paradigm in Python
CO4. apply the different File handling operations
CO5. design GUI Applications in Python
CO6. construct graphical representation of data using python packages

List of Program

---

1. Python program to take input from user and display "Hello MITS Gwalior".
2. Python program to do arithmetic operations.
3. Python program to find area of rectangle, circle and triangle.
4. Python program to check number is even or odd, prime not prime.
5. Python program find factorial of a number.
6. Python program to check year is leap year or not.
7. Python Program to implement the operation on List, Tuple, Set and Dictionary.
8. Python Program to handle the exception and file handling operation.
9. Python Program to create and use of user defined function.
10. Python Program to solve a problem using Lambda function
11. Python Program for creating an object with and without inheritance.

Skill Based Project

---

1. Visualize the Publically available real world data set using various function and identify the suitable plot for better representation.
2. Handle the missing data and categorical value in a real world Data Set.
3. Build a prediction model based on Classification Data Set.
4. Build a prediction model based on Regression Data Set.
5. Build a prediction model based on Clustering Data Set.

COMPUTER SYSTEM ORGANIZATION
(DC)(150223)

COURSE OBJECTIVES

- To provide the fundamental knowledge of a computer system and its processing units.
- To provide the details of input & output operations, memory management and performance measurement of the computer system.
- To understand how computer represents and manipulate data.

Unit -I

Introduction: Von-Neumann Model, Various Subsystems, CPU, Memory, I/O, System Bus, CPU and Memory Registers, Program Counter, Accumulator, Register Transfer and Micro Operations: Register Transfer Language, Register Transfer, Tree-State Bus Buffers, Bus and Memory Transfers, Arithmetic Micro-Operation, Logic Micro-Operation, Shift Micro- Operation Register Transfer Micro Operations, Arithmetic Micro-Operations, Logic Micro- Operations and Shift Micro-Operations.

Unit- II

Computer Arithmetic: Addition and Subtraction with Signed-Magnitude, Multiplication Algorithm, Division Algorithm, Division Algorithms, Floating-Point Arithmetic Operations.

Central Processing Unit (CPU): General Purpose Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC).Hardwired and Micro programmed Control.

Unit -III

Microprocessors: Introduction of 8085 Microprocessor: Architecture, Instruction Set, Addressing Modes, Interrupts and Basic Assembly Language Programming, Introduction to Pipelining & Multiprocessors.

Unit -IV

Input-Output Organization: Peripheral Devices, I/O Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, DMA (DMA Controller, DMA Transfer),

Input-Output Processor (IOP), Data Transfer- Serial/Parallel, Simplex/ Half Duplex/ Full Duplex.

Unit-V

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory- Organization and Mappings, Memory Management Hardware.

RECOMMENDED BOOKS

- Computer System Architecture, Morris Mano, PHI.
- Microprocessor Architecture, Programming and Applications with the 8085, Gaonkar,
- Computer Organization, Carl Hamacher, THM.
- Computer Architecture and Organization, J P Hayes, Mc-Graw Hills, New Delhi.

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. Recall the basic building blocks of computer Architecture.
CO2. Explain different memories and the functional units of a processor.
CO3. Explain the concept of working of microprocessor, multiprocessor and pipelining.
CO4. Analyze various modes of Input-Output data transfer.
CO5. Evaluate the arithmetic related to the number system.
CO6. Develop the skill of writing low level programming.

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR - 474005
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal, M.P.)

Department of Computer Science and Engineering

Department of Computer Science & Engineering

COMPUTER GRAPHICS
150224(DC)

COURSE OBJECTIVES

- To provide an introduction to the theory and practice of computer graphics.
- To give a good exposure related to Computer Graphics algorithms and to design various graphics primitives.
- To enhance the proficiency in programming skills related to animation and graphics object Design

---

Unit-I

Introduction to Computer Graphics: Interactive Computer Graphics, Application of Computer Graphics, Random and Raster Scan Displays, Storage Tube Graphics Display, Calligraphic Refresh Graphics Display, Flat Panel Display, Refreshing, Flickering, Interlacing, Resolution, Bit Depth, Aspect Ratio etc.

Unit-II

Scan Conversion Technique: Image representation, Line drawing: DDA, Bresenham's Algorithm. Circle Drawing: General Method, Mid-Point, DDA, Bresenham's Circle Generation Algorithm, And Ellipse Generation Algorithm, Curves: Parametric Function, Bezier Method, B-Spline Method.

Unit-III

2D & 3D Transformations: Translation, Rotation, Scaling, Reflection, Shearing, Inverse Transformation, Composite Transformation, World Coordinate System, Viewing Transformation, Representation of 3D object on Screen, Parallel and Perspective Projections.

Unit-IV

Clipping: Point clipping, Line Clipping, Simple Visibility Line Clipping Algorithm, Cohen Sutherland Line Clipping Algorithm etc., Polygon Clipping, Convex and Concave Polygon, Sutherland Hodgeman Polygon Clipping Algorithm etc., Area Filling, Hidden Surface Elimination: Z- Buffer algorithm and Painter's Algorithm.

Unit-V

Basic Illumination Models: Diffuse Reflection, Specular Reflection, Phong Shading, Gouraud Shading, and Color Models like RGB, YIQ, CMY, HSV etc., and Introduction to Digital Image Processing (DIP), Fundamental Steps and Components of DIP.

RECOMMENDED BOOKS

- Computer Graphics, Donald Hearn and M.P. Becker, PHI Publication.
- Computer Graphics principle and Practice, FoleyVandam, Feiner, Hughes.
- Principles of Computers Graphics, Rogers, TMH.
- Computer Graphics, Sinha and Udai, TMH.
- Digital Image Processing, Gonzalez.

COURSE OUTCOMES

After completion of the course students will be able to:

CO1. Explain interactive Computer Graphics, various display devices and explore applications of computer graphics.
CO2. Illustrate various line generations, circle generation, curve generation and shape Generation algorithms.
CO3. Apply various 2-Dimensional and 3-Dimensional transformations and projections on Images.
CO4. Classify methods of image clipping and various algorithms for Line and Polygon clipping.
CO5. Choose appropriate filling algorithms, Hidden Surface Elimination algorithm and apply on various images.
CO6. Discuss various color models, shading methods, animation and Digital Image Processing.

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR - 474005
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal, M.P.)

Department of Computer Science & Engineering

COMPUTER GRAPHICS
150224(DC)

List of Experiments

1. Installation and Introduction to OpenGL basics, graphic functions, commands for compiling and executing an OpenGL Program.
2. Write an OpenGL Program to create an output window, to plot a point with given coordinates and other basic demonstrations.
3. Write an OpenGL Program to implement DDA Line Drawing Algorithm.
4. Write an OpenGL Program to implement Bresenham Line Algorithm.
5. Write an OpenGL Program to implement Mid-Point Circle Algorithm.
6. Write an OpenGL Program to implement following 2D transformations:
    i. Translation of a point, line and polygon.
    ii. Scaling of a line and polygon.
    iii. Rotation of a line and polygon around origin.
7. Write an OpenGL Program to implement:
    i. Flood Filling Algorithm using polygon.
    ii. Boundary Filling Algorithm using polygon.

Skill Based Projects

1. Develop a project to implement a stretch band effect. In which a user will click on the screen and drag the mouse / arrow keys over the screen coordinates. The line should be updated like rubber-band and on the right-click gets fixed.
2. Develop a project to implement the DDA algorithm for drawing line. In this project a programmer is expected to shift the origin to the center of the screen and divide the screen into required quadrants.
3. Develop a project with menu option to input the line coordinates from the user to generate a line using Bresenham's method and DDA algorithm.
4. Develop a project to demonstrate 2D animation such as clock simulation.
5. Develop a project to demonstrate 2D animation such as rising sun.
6. Develop a project to implement the bouncing ball inside a defined rectangular window.
7. Develop a project to to draw Bezier and B-Spline Curves with interactive user inputs for control polygon defining the shape of the curve.
8. Develop a project to demonstrate shear transformation in different directions on a unit square situated at the origin.
9. Develop a project in which a set of lines and a rectangular area of interest is given by user, the task is to remove lines which are outside the area of interest and clip the lines which are partially inside the area.

Department of Computer Science and Engineering

Hardware and Troubleshooting Lab(150225)

COURSE OBJECTIVES:

- To understand various number systems, boolean algebra, logic Gates.
- To acquire the knowledge of a computer system, motherboard and Its processing unit.
- To be aware of different memories, I/O devices, installation and SMPS.

Unit – I:

Number System, Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Boolean Relations, Digital Logic Gates, De Morgan's Theorem, Karnaugh Maps and simplifications.

Unit – II:

Combinational Circuits. Half Adder, Full Adder, Binary Adder-Subtractor, Binary Multiplier, Comparator, Decoders, Encoders, Multiplexers.

Unit – III:

Sequential Circuits, Latches, Flip-Flops: Edge-triggered D Flip-flop, Edge-triggered JK Flip-flop, JK Master-slave Flip-flop, Registers. Integrated circuits.

Unit – IV:

Introduction of Motherboard, components of Motherboard, Types of Motherboard, Bios, Form-factor, CPU Sockets, types of memory sockets, IDE Ports, chipset, Integrated Peripherals, Peripheral card slots, Bootstrapping process.

Unit – V:

Introduction to Memory, Types of Memory, Installation and Partition of Hard Disk, Working of HDD and SDD. Basics of I/O Devices, Buffering and spooling, Introduction to Ports, Identify the Different Ports, Ports Troubleshooting. Linux and Windows Installation. SMPS (Switch Mode Power Supply).

RECOMMENDED BOOKS:

- **Digital Design, Morris Mano M. and Michael D. Ciletti, IV Edition, Pearson Education.**
- **Digital Electronics: Principles, Devices and Applications, Anil K. Maini, Wiley.**
- **The Indispensable PC Hardware Book, Hans-Peter Messmer, Third Edition.**

COURSE OUTCOMES:

After completion of this course, the students would be able to:

CO1. Illustrate the concept of number system and Boolean algebra.
CO2. Demonstrate installation of OS and connections through ports at basic level.
CO3. Build various circuits and inspect their working.
CO4. Examine the ICs specifications and their functioning.
CO5. Explain the concept of Memory, Motherboard, Bus, and SMPS.
CO6. Choose appropriate logic gates to design combinational & sequential circuits.

Hardware LAB

List of Experiments

1. Study the different parts of computer system
2. Study different parts of motherboard
3. Study various types of connectors
4. Draw the pin details of various connectors
5. Study CMOS setup
6. Partition and format the hard disc
7. Installation of OS: Linux and windows
8. Connect systems in network using switch
9. Connect the systems in peer to peer network
10. Configure e-mail server
11. Configure e-mail client
12. Configure browser for Internet access using proxy server
13. Configure Virtual Private Network (VPN)
14. Study of PC Troubleshooting.

Skill Based Project

- **To create a multiple Star topology.**
- **To connect a multiple hard disk drive in a computer and then create a multiple volume.**
- **To create multiple Wi-Fi access network.**
- **How to do complete pc assembling.**
- **To create a domain server with a client .**
- **To create a DHCP server and assign dynamic IP address in a local pc**
- **Explain in detail to install multiple windows in a pc**
- **To create pair to pair network**
- **To create combine LAN & WAN network.**
- **To describe IP address and a connect pc different network IP address.**