|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Unit** | **Topic & sub – topic** | **Topics Covered** | **COs** | **Lectures proposed** | **Lecture delivered** | **Date** | **No. of students present** **( )** | **Sign. of faculty** |
| **1** | Basic Terminology, Elementary Data Organization, Built in Data Types in C. |  | CO1 | 1 | Yes | 09th Sept | 28 |  |
| Efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations: Big Oh, Big Theta and Big Omega, Time-Space trade-off, Sparse Matrices and their representations. |  | CO1 | 1 | Yes | 10th Sept 11th Sept | 25-10th  27 -11th  |  |
| Linked lists, Abstract Data Types (ADT) Arrays: Definition, Single and Multidimensional Arrays, and Application of arrays. |  | CO1 | 1 | Yes | 13th Sept | 32 |  |
| Representation of Arrays: Row Major Order, and Column Major Order, Derivation of Index Formulae for 1-D, 2-D, 3-D and n-D Array. |  | CO1 | 1 |  |  |  |  |
| Array Implementation and Pointer Implementation of Single Linked List. |  | CO1 | 1 | Yes | 24th Sept | 50 |  |
| Singly Linked Lists, Operations on a Linked List. Insertion, Deletion and Traversal. |  | CO1 | 1 |  |  |  |  |
| Doubly Linked List- Operations on a Doubly Linked List. Insertion, Deletion and Traversal. |  | CO1 | 1 |  |  |  |  |
| Circularly Linked List- Operations on a Circular Linked List. Insertion, Deletion, Traversal and Polynomial Representation. |  | CO1 | 1 |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Total Lectures** | **8** |  |  |  |  |
| **2** | Stacks: Abstract Data Type, Primitive Stack operations: Push & Pop, Array Implementation of Stack in C. |  | CO1 | 1 | Yes | 17th  and 18th Sept |  22- 17th 34-18th  |  |
| Linked Implementation of Stack in C, Tradeoffs between iteration and recursion Dequeue and Priority Queue. |  | CO1 | 1 |  |  |  |  |
| Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression |  | CO5 | 1 | Yes | 20th Sept, 23th Sept |  26- 20th  48- 23th   |  |
| Iteration and Recursion- Principles of recursion, Tail recursion, Removal of recursion. |  | CO4 | 1 |  |  |  |  |
| Problem solving using iteration and recursion with examples such as binary search, Fibonacci numbers. |  | CO4 | 1 |  |  |  |  |
| Solving Hanoi towers. |  | CO4 | 1 |  |  |  |  |
| Queues: Operations on Queue: Create, Add, Delete, Full and Empty. |  | CO1 | 1 |  |  |  |  |
| Circular queues, Array implementation of queues in C and Linked implementation of queues in C. |  | CO1 | 1 |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Total Lectures** | **8** |  |  |  |  |
| **3** | Concept of Searching, Sequential search, Index Sequential Search.  |  | CO2 | 1 |  |  |  |  |
| Binary Search and Concept of Hashing. |  | CO2 | 1 |  |  |  |  |
| Collision resolution Techniques used in Hashing. |  | CO2 | 1 |  |  |  |  |
| Sorting: Insertion Sort, Selection. |  | CO2 | 1 |  |  |  |  |
| Bubble Sort, Quick Sort. |  | CO2 | 1 |  |  |  |  |
| Merge Sort. |  | CO2 | 1 |  |  |  |  |
| Heap Sort. |  | CO2 | 1 |  |  |  |  |
| Radix Sort. |  | CO2 | 1 |  |  |  |  |
| **Total Lectures** | **8** |  |  |  |  |
| **4** | Basic terminology used with Tree, Binary Trees, Binary Tree Representation: Array Representation and Pointer (Linked List) Representation.  |  | CO3 | 1 |  |  |  |  |
| Strictly Binary Tree, Complete Binary Tree. A Extended Binary Trees. |  | CO3 | 1 |  |  |  |  |
| Binary Search Tree and Tree Traversal algorithms: Inorder, Preorder and Postorder. |  | CO3 | 1 |  |  |  |  |
| Constructing Binary Tree from given Tree Traversal, Operation of Insertion in Binary search. |  | CO3 | 1 |  |  |  |  |
| Operation of Deletion, Searching & Modification of data in Binary Search. |  | CO3 | 1 |  |  |  |  |
| Threaded Binary trees, Traversing Threaded Binary trees. |  | CO3 | 1 |  |  |  |  |
| Huffman coding using Binary Tree & Binary Heaps. |  | CO3,CO5 | 1 |  |  |  |  |
| Concept & Basic Operations for AVL Tree and B Tree.  |  | CO3,CO5 | 1 |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Total Lectures** | **8** |  |  |  |  |
| **5** | Graphs: Terminology used with Graph.  |  | CO3 | 1 |  |  |  |  |
| Representations: Adjacency Matrices, Adjacency List, Adjacency. |  | CO3 | 1 |  |  |  |  |
| Data Structure for Graph. |  | CO3 | 1 |  |  |  |  |
| Graph Traversal: Depth First Search and Breadth First Search, Connected Component |  | CO3 | 1 |  |  |  |  |
| Minimum Cost Spanning Trees: Kruskal algorithm |  | CO3 | 1 |  |  |  |  |
| Spanning Trees, Minimum Cost Spanning Trees: Prims algorithm |  | CO3 | 1 |  |  |  |  |
| Transitive Closure and Shortest Path algorithm: Warshal Algorithm |  | CO3 | 1 |  |  |  |  |
| Transitive Closure and Shortest Path algorithm: Dijikstra Algorithm. |  | CO3,CO5 | 1 |  |  |  |  |
| **Total Lectures** | **8** |  |  |  |  |

**BBS College of Engineering and Technology Prayagraj**

**Lecture Plan**

|  |  |
| --- | --- |
| Department |  Computer Science Engineering |
| Program/Year/Semester/Sec | B Tech CS 2nd Year |
| Course Name/ Course Title | Data Structures |
| Course Code | BCS 301 |
| Name of Faculty  | Ms Shreya Agarwal |
| Department of Faculty  | Computer Science (BBS College of Engineering and Technology) |

|  |  |
| --- | --- |
| Pre-requisites for the Course | A Solid grasp of a programming language like C/C++. |
| Understanding how to declare, assign, and manipulate variables of different data types (e.g., integers, floating-point numbers, strings, Booleans). |
|  | Familiarity with conditional statements (if-else) and loops (for, while) to control the execution of code.  |
|  |  Knowledge of creating and using functions to modularize code and improve readability. |
|  |  Understanding simple algorithms like searching and sorting. Ability to think logically and reason about problem-solving. |

|  |  |
| --- | --- |
| Type of Course | Theory/Lecture |
| Contact hours | 40 hrs |

|  |
| --- |
| **Course Outcomes (COs)** |
| At the end of this course students will demonstrate the ability to: |
| CO1 | Describe how arrays, linked lists, stacks, queues, trees, and graphs are represented in memory, used by the algorithms and their common applications. |
| CO2 | Discuss the computational efficiency of the sorting and searching algorithms. |
| CO3 | Implementation of Trees and Graphs and perform various operations on these data structure.  |
| CO4 | Understanding the concept of recursion, application of recursion and its implementation and removal of recursion. |
| CO5 | Identify the alternative implementations of data structures with respect to its performance to solve a real world problem. |

|  |
| --- |
| **Text Books & References** |
| 1 | Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, “Data Structures Using C and C++”, PHI Learning Private Limited, Delhi India. |
| 2 | Lipschutz, “Data Structures” Schaum’s Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd. |
| 3 | Jean Paul Trembley and Paul G. Sorenson, “An Introduction to Data Structures with applications”, McGraw Hill. |
| 4 | R. Kruse et. al, “Data Structures and Program Design in C”, Pearson Education. |
| 5 | AK Sharma, “Data Structure Using C”, Pearson Education India. |

 Signature of Faculty Signature of HOD

|  |  |
| --- | --- |
|  **Comments** |  |