**University of Central Punjab**

**Faculty of Information Technology**



**BSSE**

**PROGRAM (S) TO BE**

**EVALUATED**

1. **Course Description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Code** | SECP2043 | | | |
| **Course Title** | Data Structures and Algorithms | | | |
| **Credit Hours** | 3+1 | | | |
| **Prerequisites** | Students are expected to have the following background:   * Reasonable programming skills with proven track in ITC, PF and OOP courses. | | | |
| **Assessment Instruments with Weights** (homework, quizzes, midterms, final, programming assignments, lab work, etc.) | |  |  |  |  | | --- | --- | --- | --- | | **Theory** | | | | | **Sr.No.** | **Assessment Type** | **%age** | **Count** | | 1 | Quiz | 10 | 4 | | 2 | Assignments | 15 | 4 | | 3 | Class Participation | 10 | Instructor’s Choice | | 4 | Midterm | 20 | 1 | | 5 | Final Term | 45 | 1 |  |  |  |  |  | | --- | --- | --- | --- | | **Lab** | | | | | **Sr.No.** | **Assessment Type** | **%age** | **Count** | | 1 | Class Participation | 5 | Instructor’s Choice | | 2 | Graded Labs | 20 | 4 | | 3 | Project | 10 | 1 | | 4 | Midterm | 20 | 1 | | 5 | Final Term | 45 | 1 | | | | |
| **Course Instructor** | Afifa Hameed | | | |
| **Course Coordinator** | Dr. Nabeel Sabir Khan | | | |
| **Office Hours** |  | | | |
| **Plagiarism Policy** | All the parties involved in first cheating case will be awarded Zero for that evaluation. Afterwards, if any cheating made by any person of the class, will get F in course for all the accused parties. | | | |
| **Course Tools** | DSA RoadMap: <https://bit.ly/3dyWMmH>  Data Structures Visualization: <https://bit.ly/3w6ftEq>  VisualGo for Graphs Visualization: <https://bit.ly/3dxSonX>  Sorting Algorithms Visualization: <https://bit.ly/3T5yXTV> | | | |
| **Course Description** | This course familiarizes students with the concepts of creating, storing, retrieving, ordering, and manipulating data structures, along with the fundamentals of algorithm analysis. Students will gain an in-depth understanding of the formal specification of data structures and their real-world applications. These goals will be accomplished by:   * Introducing students to fundamental data structures and their associated algorithms * Exploring the theory of complexity and developing skills to analyze time and space requirements for different data structures and algorithms * Implementing data structures in Java using both custom implementations and built-in libraries such as ArrayList, LinkedList, HashMap, TreeSet, and others * Understanding the appropriate use of different data structures in various scenarios to optimize performance and efficiency. | | | |
| **Course Objectives** | On completion of this module, students should be able to:   1. Understand the properties of various data structures. 2. Identify the strengths and weaknesses of different data structures. 3. Design and employ appropriate data structures for solving computing problems 4. Possess the knowledge of various existing algorithms. 5. Analyze and compare the efficiency of algorithms. | | | |
| **Textbook** | **Text Book** (Any of the following)**:**  Data Structure and Algorithms in Java by Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser 6th Edition  Data Structure and Algorithms in Java, Second Edition by Robert Lafore | | | |
| **Reference Material** | **Reference:**  Data Structures and Algorithm Analysis in Java– Mark Allen Weiss  **Web References:**   * GeeksforGeeks - DSA in Java   <https://www.geeksforgeeks.org/data-structures/>   * JavaTpoint - Data Structures in Java   <https://www.javatpoint.com/data-structure-tutorial>   * TutorialsPoint - Java Data Structures   <https://www.tutorialspoint.com/data_structures_algorithms/index.htm>   * LeetCode - Java Solutions for DSA Problems   <https://leetcode.com/> | | | |
| **Topics Covered in the Course, with Number of Lectures on Each Topic** (assume 15-week instruction and one-hour lectures) | Attached | | | |
| **Programming Assignments Done in the Course** | 4-5 Assignments  1 Term Project | | | |
| **Class Time Spent on** (in credit hours) | **Theory** | **Problem Analysis** | **Solution Design** | **Social and Ethical Issues** |
| 2 | ½ | ½ | 0 |
| **Oral and Written Communications** |  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLO** | **CLO STATEMENT** | **Level** | **PLO** |
| 1 | **Describe** the characteristics and use cases of various data structures. | C2 | 2 |
| 2 | **Examine** the time and space complexity of different data structures and algorithms. | C4 | 3 |
| 3 | **Implement** a range of data structures and algorithms. | C3 | 4 |
| **Lab** | | | |
| 1 | **Demonstrate** proficiency in implementing a variety of data structures and algorithms in Java to solve complex computational problems. | C3 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Week #** | **Topics Covered** | **Book Reference** | **Evaluation Instrument Used** |
| 1 | **Introduction to the Data Structure and Algorithms and Java Collection Framework**   * Applications of data structure and algorithm * Types of DS, Big-O notation * Difference between Best, Worst and Average Case Analysis * Java Collections (HashMap, PriorityQueue, ArrayList, LinkedList)   Practical Application:  **Online Ticket Booking** (Dynamic seating with ArrayList, queue management with LinkedList) |  |  |
| 2 | **Introduction to List, and Interfaces, Basics of Arrays and ArrayLists**   * Common Operations and their complexity analysis such as insert and delete * Common Array operations such as traverse and search * Static & dynamic arrays   Practical Applications:  **E-commerce Inventory System** (Using ArrayList for product management) |  |  |
| 3 | **Introduction to Linked List, Linked List vs. Array List**   * Singly Linked List, insertion, deletion, traversal, searching * Complexity analysis of singly linked list operations   Practical Applications:  **Music Playlist Management** (Forward & backward navigation) |  |  |
| 4 | **Type of Linked List (Doubly, Circular)**   * Doubly Linked List, Circular Linked List, Java’s LinkedList class, complexity analysis * Comparison of Singly with Circular Linked List in terms of complexity * Analyzing reversal, merging and detecting cycles in linked list   Practical Applications:  **Browser History Navigation System** (Back and forward functionality) |  |  |
| 5 | **Introduction to Stacks (Array-based & Linked List-based)**   * Using Stack Class in Java * LIFO principle, stack operations (push, pop, peek), Java’s Stack<E> and Deque<E>   Practical Applications:  **Undo/Redo System in Text Editors, Balanced Parenthesis, Expression Evaluation** |  |  |
| 6 | **Introduction to Queues in Java (Array-based & Linked List-based)**   * FIFO principle, types of queues (PriorityQueue, Deque) * Types of Queues – Circular and Non-Circular   Practical Applications:  **Task Scheduling System** (Handling task priorities with queues) |  |  |
| 7 | **Recursion in Java**   * Recursive vs. iterative approach * Using Recursion and discussing stack overflow * Base & recursive cases, factorial, Fibonacci, Tower of Hanoi   Practical Applications:  **Recursive Binary Search** on a large dataset**,** Maze Solving, Sudoko Puzzle |  |  |
| 8 | Revision Week  (Consolidation of Linked Lists, Stacks, Queues, and Recursion) |  |  |
| 9 | **Trees - Introduction & Terminology**   * Trees and traversal algorithm, pre-order, postorder and inorder traversal * Binary Trees   Practical Applications:  **File System Representation** (Tree-based directory structure)  **Introduction to BST**   * Binary Search Tree operations (insert, delete, search) * Using Java’s TreeSet and TreeMap |  |  |
| 10 | Practical Applications:  **Dictionary/Spell Checker** (Fast word lookups with BST)  **Introduction to Advanced Trees**   * Advanced Trees: AVL Tree or Red Black Tree * Common Operations of AVL Tree |  |  |
| 11 | **Introduction to Tries data structure**   * prefix searches, autocomplete   Practical Applications:  **Auto-completion in Search Engines** (Efficient word lookups using Tries)  **Recursion with Trees**  **Expression Tree Evaluation** (Recursive traversal to evaluate expressions) |  |  |
| 12 | **Introduction to Hashing and Hash Table**   * Hash functions, collision resolution (chaining, probing) * Java’s HashMap, HashSet, vs. TreeMap * Hash table operations, time and space complexity   Practical Applications:  **Fast Duplicate Detection, LRU Cache Implementation** |  |  |
| 13 | **Introduction to Graphs representations and complexity**   * Graph representation (adjacency matrix, adjacency list) * Directed & undirected graphs   Practical Applications:  **Social Network Graph** (Friend connections & mutual suggestions using BFS/DFS)  **Graph Traversal Algorithms**   * BFS (queue-based), DFS (stack-based & recursive), shortest path algorithms   Practical Applications:  **Network Routing Algorithm** (Finding shortest paths in networks) |  |  |
| 14 | **Heaps & Priority Queues**   * Min-Heap, Max-Heap * Java’s PriorityQueue, HeapSort   Practical Applications:  **Task Scheduling System** (Prioritizing tasks with Min-Heap) |  |  |
| 15 | **Introduction to Sorting Algorithms**   * Bubble Sort, Selection Sort, Insertion Sort * Merge Sort, Quick Sort, Java’s sorting internals   Practical Applications:  **Large-Scale Data Sorting** (Efficient sorting for huge datasets) |  |  |
|  |
| 16 | Revision Week |  |  |