|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **The Superior University Lahore**  **Faculty of Computer Science & Information Technology**  **Course Outline** | | | | | | | | | | | | | | |
| Course Information | Course Title | | **Data Structure & Algorithm (Python)** | | | | | | | | | | | |
| Course ID | |  | | | | Course Type | | | Computing Core | | | | |
| Credit hours | | 3 + 1 | | | | Hours per week (C-L) | | | 3 + 1 | | | | |
| Programs | | BS (AI), BS (DS) | | | | Preferred Semester | | | 4 | | | | |
| Date | | 2025-01-06 | | | | Version | | | 1.0 | | | | |
| Instructor | | Hafiz Muhammad Tayyab Khushi | | | | TA / Lab Engineer | | | Danial Qamar | | | | |
| Course Description | The purpose of this course is to provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. The main objective of the course is to teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter. This course is also about showing the correctness of algorithms and studying their computational complexities. This course offers the students a mixture of theoretical knowledge and practical experience.  The study of data structures and algorithms is carried out within an object-oriented framework. When implementations are considered, the Python programming language is used.  The course focuses on basic and essential topics in data structures, including array-based lists, linked lists Introduction to Data Structures and Algorithms, Complexity Analysis,  Arrays, Sorting Algorithms: Insertion Sort, Selection Sort, Bubble Sort, Shell Sort, Heap Sort, Quick Sort, Merge Sort, Radix Sort, Bucket Sort, Linked Lists: Singly Linked Lists, Doubly Linked Lists, Circular List, Stacks, Queues, and Priority Queue, Recursion: Function call and Recursion Implementation, Tail Recursion, Non-tail Recursion, Indirect Recursion, Nested Recursion, Backtracking, Trees: Binary Trees, Binary Heap, Binary Search. Tree Traversal, Insertion, Deletion, and Balancing a Tree, Heap, B-Tree, Spanning Tree, Splay Trees, Graphs: Representation, Traversal, Shortest Path, and Cycle Detection, Isomorphic Graphs; Graph Traversal Algorithms, Hashing, Memory Management and Garbage Collection. | | | | | | | | | | | | | |
| Course Objectives | *The objective of this course is to enable students;* | | | | | | | | | | | | | |
| No. | Objective | | | | | | | | | | | | Relation with Program  Objectives |
|  | To impart the basic concepts of data structures and algorithms. | | | | | | | | | | | | POL 1,2,3 |
|  | To understand concepts about searching and sorting techniques | | | | | | | | | | | | POL 4 |
|  | To Understand basic concepts about stacks, queues, lists, trees and graphs. | | | | | | | | | | | | POL 1-4 |
| 4. | To understanding about writing algorithms and step by step approach in solving problems with the help of fundamental data structures | | | | | | | | | | | | POL 1-4 |
| Course Learning Outcomes (CLO) | *At the end of this course students will be able to demonstrate;* | | | | | | | | | | | | | |
| No. | Outcome | | | | | | | | | | | | Relation with PLO/PLO |
|  | Explain the need for efficiency in data structures and algorithms. | | | | | | | | | | | | PLO 1 |
|  | Apply methods to analyze running time of essential data structures and estimate efficiency of the algorithms and implementations. | | | | | | | | | | | | PLO 2 |
|  | Understand and apply the concept of abstract data type to represent and implement heterogeneous data structures. | | | | | | | | | | | | PLO 2 |
|  | Write programs using array-based lists, write programs using linked lists, write programs that use skiplists. write code for hash tables, and compare and contrast various collision detection and avoidance techniques. | | | | | | | | | | | | PLO 3 |
|  | Demonstrate skills in tracing, analyzing, and designing recursive algorithms and recursive methods. | | | | | | | | | | | | PLO 3 |
|  |  | Write programs using binary trees and variations. | | | | | | | | | | | | PLO 1-3 |
|  |  | Analyze and implement different types of sorting algorithms. | | | | | | | | | | | | PLO 1-3 |
|  |  | Implement data structures for graphs and approaches for searching graphs using breadth-first, depth-first, best-first search, etc. | | | | | | | | | | | | PLO 1-4 |
| Lecture type |  | | | | | | | | | | | | | |
| Prerequisites | Object Oriented Programming | | | | | | | | | | | | | |
| Follow up  Courses | Design & Analysis of Algorithm | | | | | | | | | | | | | |
| Course Software  or Tool | **PyCharm (Recommended)**, Jupyter Notebook, Anaconda, Spyder, Visual Studio Code | | | | | | | | | | | | | |
| Textbook | Title | | | | Edition | Authors | | | Publisher | | Year | | ISBN | |
| Data Structures and Algorithms in Python, | | | | Anu | Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser | | | Wily | | 2014 | | 978-0-13-284737-7 | |
| Data Structures and Algorithms using Python | | | | 4th | Rance D Necaise | | | Wily | | 2016 | | 1-133-60842-6 | |
| Assessment  Criteria (100%) | Assessment | | Weight | Used to attain CLO | | | | Assessment | | | | Weight | Used to attain CLO | |
| Assignment | | 10% |  | | | | Quiz | | | | 10% |  | |
| Lab | | 0% |  | | | | Project / Presentation | | | | 20% |  | |
| Attendance | | 0% |  | | | | Participation | | | | 0% |  | |
| Mid Term | | 20% |  | | | | Final | | | | 40% |  | |
| Methods of  Evaluation | Quizzes, Assignments, Mid Term, Final Term, and Semester Project. | | | | | | | | | | | | | |
| Notes |  | | | | | | | | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week No.** | **Topic** | **Lecture No.** | **Lecture Contents** | **Relation with CLO** |
|  | **Introduction to Data structures and algorithms (DSA)** |  | * Introduction to data structures and algorithms * Why do we need data structures and algorithms? * Course module and book discussion | CLO1 |
|  | * Discussion on different data structures * Discussion on different algorithms | CLO1 |
|  | **Stack & Queue**  **Python Implementation** |  | * Stack | CLO1 |
|  | * Queue |
|  | **Recursions &**  **Sorting Algorithms**  **Python Implementations** |  | * Concept of recursion * Insertion Sort * Selection Sort * Shell Sort Heap Sort * Bubble Sort * Quick Sort * Merge Sort * Radix Sort * Bucket Sort | CLO 1,2,3 |
|  |
|  |  |
|  |
|  | **Linked List**  **(Implementation using python list)** |  | * Singly Linked List * Insertion   + Beginning of List.   + End of List. * Before / After any number / Location * Deletion   + Beginning of List.   + End of List.   + Before / After any number / Location | CLO2 |
|  | * Searching   + Beginning of List.   + End of List.   + Before / After any number / Location |
|  | **Double & Circular Linked List** |  | * Double & Circular Linked (Insertion) |
|  | * Double & Circular Linked List (Deletion & Searching) |
|  | **NUMPY ARRAYS Data Structure** |  | * Difference between python list and numpy * Mathemtical operations * Martix operations * N Dimentionla Array Opearations |
|  | * All sorting algorithems using the NUMPY arrays and measuring the and time and space complexity using the Python functions | CLO 2,3 |
|  | **Python PANDAS Data Structures – 1** |  | * Difference between the python list and Pandas Series & Pandas Data Frames * Difference between NUMPY List and Pandas data frames |
|  | * Revision |
|  | **Mid Term** |  | * Mid Term Week |  |
|  |
|  | **Python PANDAS Data Structures- 2** |  | * Reading the Large data sets and csv files using the pandas data frames | CLO 1,2,3 |
|  | * Row / column / matrix and n-dimensional operations using the combinations of pandas data structure and NumPy operations |
|  | **Tree**  **with NUMPY Arrays, Pandas and Normal python Lists** |  | * Trees with python list | CLO 1,2,3,4 |
|  | * Tree with numpy arrays and pandas series |
|  |  | * Tree & its Terminology | CLO 1,2,3,4 |
|  | * Binary Tree * Binary Search Tree * AVL Tree * Tree Balancing |
|  | **Tree Implementation** |  | * Implementation of Binary Tree | CLO 1,2,3,4 |
|  | * Tree Traversing |
|  | **Graphs**  **(Python Implementations) with NUMPY Arrays & Normal Lists** |  | * Introduction of Graphs | CLO 1,2,3,4 |
|  | * Graph Types |
|  |  | * Graph Traversing |
|  |
|  | **Hashing** |  | * Memory Management | CLO 1,2,3,4 |
|  | * Garbage Collection |
|  | **Final Exam** |  | * Final Exam |  |