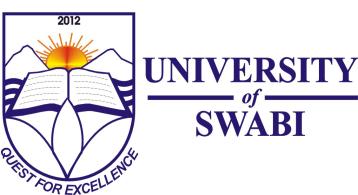
**UNIVERSITY OF SWABI **

**DEPARTMENT OF COMPUTER SCIENCE**

**Course Outlines**

**Fall Semester 2023**

|  |  |
| --- | --- |
| **Course Code** |  |
| **Course Title** | Data Structures and Algorithms |
| **Credit Hours** | 4 (3+1) |
| **Prerequisites by Course(s) and Topics** | Object-Oriented Programming |
| **Assessment with Weights** | |  |  |  | | --- | --- | --- | | Assignments | : | 5% | | Test/Quiz | : | 10% | | Midterms | : | 20% | | Final  Lab : | : | 35%  30 | |
| **Teacher Name** | Mr. Muhammad khan |
| Current Catalog Description | Data Structure is a core curriculum of computing and engineering programs, which aims to cultivate students’ abilities of selecting appropriate data structure to develop efficient programs. 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 intention 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. |
| Objectives | The course is designed to teach students structures and schemes, which allow them to write programs to efficiently manipulate, store, and retrieve data. Students are exposed to the concepts of linear and nonlinear data structure, time and space complexity of computer programs. Different sorting algorithms will be examined. Important elements in the course include recursive programming as well as evaluation and description of program efficiency. |
| **Learning Outcomes** | At the end of the course students will get the following abilities:   * Able to design appropriate data structure for any give problem/program * Able to solve practical and complex problems with efficient programs * Able to analysis program efficiency using time and space parameters   Able to implement and manipulate various algorithms. |
| **Textbook/Reference Material** | 1. Data Structures and Algorithms in C++ by Adam Drozdek  2. Data Structures and Algorithm Analysis in Java by Mark A. Weiss  3. Data Structures and Abstractions with Java by Frank M. Carrano & Timothy M. Henry  4. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss  5. Java Software Structures: Designing and Using Data Structures by John Lewis and Joseph Chase |

**Major topics to be covered:**

1. Defining Data Structure, overview programming principle in C++: topics - arrays, functions, string, files, structures, structure with in structure, templates.
2. Introduction to object oriented programming: classes, inheritance, multiple inheritance, etc.
3. Abstract data types and basic concept of data structure: linear and nonlinear
4. Efficient Memory Allocation and list processing
5. Recursions, writing program using recursive techniques
6. The stack: definition, operations, implementation, and applications.
7. The Queues: Definition, operations, implementation and applications
8. The Tree: terminology, operations, implementation, and applications.
9. The Graph: terminology, operations, implementation, and applications: algorithms to traverse graph data structure
10. Basic concept of Algorithms: Searching and Sorting
11. Complexity of Algorithms and efficiency of programs.

**Week-wise distribution of course contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Week**  **No** | **Activity** | **Course Topic** | **Details of Topics** |
|
| 1 | Class Work | Introduction | * Introduction to course * What is Data Structure? * Types of Data Structure. * Basic Operation on Data Structure * What is Algorithm? * Algorithm representation. * . |
| 2-3 | Class Work | Arrays & Lists | * Overview programming principle in C++: topics - arrays, functions, string, files, structures, structure with in structure, templates * Introduction to object oriented programming: classes, & inheritance * One dimensional array with examples. * Two dimensional array with examples * Insert Item Array, * Delete Item from Array, * Efficient Memory Allocation and list processing * Recursions, writing program using recursive techniques |
| 4-5 | Class Work | Link List; | * Introducing List data structure: Array based list, Pointer based list, Linked list: Creating, Insertion, Deletion, Variation in Linked List: Doubly linked list, Header node, Applications of Linked-List * Link List Memory representation. * Operation and Implementation of Linked List   + Traverse a Linked List,   + Insert item into Linked List   + Delete item from Linked List   + Search an Item in Linked List |
| 6 | Class Work | Stacks; | * Definition, operations, implementation, and applications * Push Item into Stack, Pop Item from Stack. * Transform Infix Expression into Postfix Expression using Stack, Evaluate Postfix Expression using Stack. |
| 7 | Class Work | Queue; | * Definition, operations, implementation and applications * Insert Item into Queue, Delete Item from Queue Implementation of Queue. |
| 8 | Class Work | Recursion & Sorting Algorithm | * Recursion – revisited with application. * Factorial method example of recursion. * Sorting Techniques (Bubble, Insertion, Selection) |
| 9 | Mid Term  Exam |  | * Paper Setting and marking |
| 10-11 | Class Work | Trees; | * Tree representation. * Binary tree. * Operation on tree. |
| 12-13 | Class Work | Algorithms | * Linear Search algorithm * Binary Search algorithm * Sorting Algorithms –   Divide and Conquer: What is divide and conquer strategy. Implementation of divide and conquer strategy in Merge sort.   * Algorithm Complexity, Big O notation, etc. |
| 14-15 | Class Work | Graph Data Structure | * Representation of Graph & Terminologies. * Adjacency matrix. * Operations on Graph. |
| 16 | Class Work | Hashing | * What is Hashing and Hash Function. * Hashing Method. |
| 17 | Course Project | Project; | * Course project submission * presentation * Lab viva * Revision of major concept along preparation for exam |
| 18 | Final Term  Exam |  | Paper Setting and marking |

**Attendance Requirements**

Students are encouraged and expected to attend all lectures, or any other activity related to the course. Moreover, students are responsible for their attendance and they must meet the minimum attendance requirement policy of the Department for appearing in final term exam.

75% attendance is mandatory. Latecomers will be marked as absent.

**Laboratory Projects/Experiments**

Students will be provided manual to perform the lab work during the course. A total of 12-13 Labs/experiments have been designed for the course of BCS122 Data Structure and Algorithms. The course demands 100% attendance in the lab work. In case any student missed any lab will be required to repeat it by him/her an extra time. A 2-3 hrs Lab session will be required for each of the experiment listed below.

|  |  |  |
| --- | --- | --- |
| **Lab#** | **Title and Description of Lab** | **Method for the conduct of Lab** |
| 1 | Lab #1: Programming Principles – Input & Output and Functions: (Week no. 1) | General programming language will be used to implement this lab. |
| 2 | Lab #2 (continuation to Lab # 1): Programming Principles – Pointers, arrays and more advanced programming techniques) | General programming language will be used to implement this lab. |
| 3 | Lab #3 (continuation to Lab # 1 & 2): Programming Principles – classes - Pointers, arrays and more advanced programming techniques) | General programming language will be used to implement this lab. |
| 4 | Lab #4 Linked List lab  List Processing and pointers to structure | General programming language will be used to implement this lab. |
| 5 | Stack Lab  Lab #6 Stack – ADT and application | Problem designing followed by practical work in general programming lab. |
| 6 | Stack Lab Using linked list  Lab #7 Continuation to the Lab # 6 – and implementation of Stack – ADT using Linked List . | Problem designing followed by practical work in general programming lab. |
| 7 | Queues Lab  Lab #8 implementation of QUEUE– ADT and Application | Problem designing followed by practical work in general programming lab. |
| 8 | Link List Queues practical lab  Lab #8 Continuation of the Lab# 7 - implementation of QUEUE– ADT and Application | Problem designing followed by practical work in general programming lab. |
| 9 | Tree Lab  Lab #9 Tree – ADT and Application | Problem designing followed by practical work in general programming lab.   * Course project to be assigned. |
| 10 | Tree Lab  Lab #10 Continuation of the Lab# 9 - implementation of Tree– ADT and Application | Problem designing followed by practical work in general programming lab.   * Work on course project |
| 11 | Algorithm Lab  Lab #11 Algorithms: Searching & Sorting | Problem designing followed by practical work in general programming lab.   * Work on course project |
| 12 | Lab Project (finalization)  Lab #12 Presentation of Lab Project | Problem designing followed by practical work in general programming lab.   * Work on course project |
| 13 | Review Lab and Lab Test | Problem designing followed by practical work in general programming lab.   * Work on course project |