|  |
| --- |
| **Department of Computer Sciences**  ***Iqra National University***  ***Peshawar, Pakistan*** |



**Data Structures & Algorithms Credit Hrs: 3+1**

**Teaching Faculty: *Muhammad Adil (Assistant Professor)***

**Course 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 time and space complexity of computer programs.

**Learning Outcomes:**

After completing this course the student must demonstrate the knowledge and ability to:

1. Independently understand basic Data Structures Concepts.

2. Understand and explain Data Structures and its terminology.

3. Identify the different types of Problems and will be able to design Algorithms for them.

4. Able to solve and program for Sorting and Searching types of important issues.

5. Identify the different types of Data Structures such as Arrays, Stack, Queues and Linked Lists.

**Course Contents:**

Introduction to data structures; Basic Terminologies; Arrays, Types Logical details; Hashing Technique; Searching Techniques, Sorting Techniques; Stacks and its Logical Details; Queues, Types and their Logical Details; Trees and Terminologies; Tree Traversals; Binary Tree Traversal; Mathematical Expressions and Binary Trees Syntax Analysis;

**Text Book:**

***Data Structures; Schaum Series*:** by John R. Hubbard. McGraw Hills Publications

**Reference Book:**

***Algorithms in C++, Parts 1-4*:** Fundamentals, Data Structure, Sorting, Searching, 3/E **Robert Sedgewick**, *Princeton University* ISBN-10: 0201350882 ISBN-13: 9780201350883 Publisher: Addison-Wesley Professional

**Prerequisite:**

Students must be familiar with the fundamental programming. Students should be able to program in C/C++ or java.

Prerequisite Subject is Computer Programming-I and Object Oriented Programming.

**Grading Policy:**

|  |  |
| --- | --- |
| **Mid-Term Exam** | **30 Marks** |
| **Assignments** | **10 Marks** |
| **Quizzes/ Tests/ Presentations** | **10 Marks** |
| **Final Term Exam** | **50 Marks** |

**Syllabus and Schedule:**

|  |  |  |
| --- | --- | --- |
| **Week** | **Topics** | **Activities** |
| 1 | Introduction to D. S. & Algo:   * Basic Terminologies * Basic Operations |  |
| 2 | Introduction to Arrays   * Introduction to 1-Dim Arrays * Dop-Vector Method for 1-Dim Array |  |
| 3 | * Traversng Algo: * Insertion Algo: * Deletion Algo | Programming Assignment |
| 4 | 2-Dimensional Arrays   * Dop-Vector Method for 2-D Arrays |  |
| 5 | Different Accessing Techniques   1. Access Table 2. Hashing Technique (Bucketing) | Quiz |
| 6 | Introduction to Sorting   1. Direct / Selection Sort Technique 2. Bubble Sort Technique | Home Work Assignment / Programming Assignment |
| 7 | 1. Insertion Sort Technique 2. Quick Sort Technique 3. Merge Sort | Programming Assignment |
| 8 | Introduction to Searching   1. Sequential / Linear Searching  * General Algo: for Linear Search * Linear search Algo for 1-Dim Arrays | Programming Assignment |
|  | **Mid Semester Exam** |  |
| 9 | Binary Search   * Example on Binary Search * Algo: for Binary Search | Programming Assignment |
| 10 | Introduction to Sequential Lists   * Stacks * Push () & Pop () in Stacks |  |
| 11 | Queues   * Insertion Algo: over Queues * Deletion Algo: over Queues   De-Queues   * Circular Queues | Programming Assignment |
| 12 | Introduction to Trees   * Basic Terminologies | Quiz |
| 13 | Tree Traversing   * Level by level Traversing * Pre-Order Traversing * Post-Order Traversing | Home Work Assignment |
| 14 | Introduction to Binary Trees   * In-Order Traversing * Pre-Order Traversing * Post-Order Traversing   Construction of a Binary Tree of Given List of Numbers | Home Work Assignment |
| 15 | Construction of a Binary Tree of Given Expression  Syntax Analysis   * In-Fix Notation * Pre-Fix Notation * Post-Fix Notation | Home Work Assignment |
| 16 | Linked List   * One Way Linked List |  |
|  | **Final Semester Exams** |  |

**Technology Involved:**

As before studying Data Structures the students have already learned to write programs in different Programming Languages like C, C++ or VB or what so ever. Therefore, I don’t have any reservations on any specific language. Most of the Assignments in this course are to convert the algorithms being discussed in the lectures in some computer programs, so a student is freely allowed writing the program in his/her choice of language.

**Note:** Students are strictly prohibited to make use of the library functions available for the discussed topics e.g. BinarySearch() in a language can’t strictly be used in a program to explore the Binary Search algorithm.

**Techniques Practiced:**

After discussing a topic it will fully be explained on White Board. The computational details and numerical solutions will be solved. After understanding all the details the algorithm for the topic will be discussed and be executed.

Students will be required to re execute the technique on some unique examples in the class. After completing the topic completely the students will be given an assignment to convert the algorithm in a Computer Program.

**Analysis of Effectiveness of Course:**

1. **Technology:** As discussed earlier the students are free to use any sort of programming language. Data Structures is a general subject and the techniques being learned in this course are not software dependent.
2. **Emerging Development Paradigms:** Most assembly languages and some low-level languages, such as BCPL (Basic Combined Programming Language), lack built-in support for data structures. On the other hand, many high-level programming languages and some higher-level assembly languages, such as MASM, have special syntax or other built-in support for certain data structures, such as records and arrays. For example, the C and Pascal languages support structs and records, respectively, in addition to vectors (one-dimensional arrays) and multi-dimensional arrays.

Modern languages also generally support modular programming, the separation between the interface of a library module and its implementation. Some provide opaque data types that allow clients to hide implementation details. Object-oriented programming languages, such as C++, Java and Smalltalk may use classes for this purpose.

1. **Pertaining to Industry:** In interviews most of the big companies like Microsoft, Oracle and Amazon etc. mainly focus on data structures. Companies usually focus on the intricate aspects of Data Structures like pointers based memory manipulation and time complexities. These companies even drill down to the bit and byte details. Therefore, to survive in a real Computer’s world it is good to fall back upon Data Structures.