**COURSE OUTLINES**

**INSTITUTION**  **DCS&SE, FBAS, International Islamic University, Islamabad.**

**Course Title: CS 214 Data Structures and Algorithms – 4 Cr. Hrs**

**Course Instructor: Dr. M. Nadeem, Assistant Professor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class:** | BSCS-F19 & BSSE-F19 | **Semester:** | 4th |
| **Instructor**  **Email:** | [nadeem@iiu.edu.pk](mailto:nadeem@iiu.edu.pk) | **Office:** | Room-105 |
| **Counseling Hours:** | See timetable | | |
| **Pre-requisites:** | Programming Fundamental  Object Oriented Programming | | |
| **Course Description:** | Abstract data types (ADTs), vector, list, deque, stack, queue, graph, digraph, table, map (associative array), priority queue, sets, trees etc. Efficient program design requires good matching of data structures (which determine how the data can be easily accessed and manipulated) and algorithms (strategies for processing the data to achieve the desired program goals). Algorithm design, complexity analysis and correctness proof form important components in study of algorithms. | | |
| **Course Objectives:** | The objective of the course is to teach students how to design, write, and analyze the performance of C/C++ programs that handle structured data and perform more complex tasks, typical of larger software projects. Students should acquire skills in using generic principles for data representation & manipulation with a view for efficiency, maintainability, and code-reuse. Successful students will, at the end of the course, be able to demonstrate analytical comprehension of concepts such as abstract data types (vectors, link lists, deques, trees, etc.), generic programming techniques (containers, adaptors, accessing data through interface, iterators, etc.), algorithms (sorting, using stacks and queues, tree exploration algorithms, etc.), and efficiency analysis (which data structures allow efficient interfaces to particular forms of data access, such as random vs. sequential data access or insertion). The students should be able to demonstrate similar skills in related implementation tasks in the C/C++ language, including extensive use of templates to allow for modularity and re-usability of code. | | |
| **Course Outcomes:** | Upon successful completion of this course, the student will be able to:  (Knowledge based)   * Competently program in C/C++ in the OO paradigm, * Manage memory usage in C/C++ programs, * Explain fundamental computing algorithms, * Analyze algorithms and identify key algorithmic strategies, * Demonstrate knowledge of programming language design issues, * Use abstract data types to help solve programming problems * Work competently in a group to learn software concepts. | | |
| **Books:** | **REQUIRED TEXTS:**   * C++ Plus Data Structure by Nell Dale, 3rd Edition * Data Structures & Algorithms in Java by M. T. Goodrich 6th Edition   **REFERENCES:**   * A Practical Introduction to Data Structures and Algorithm Analysis C. A. Shaffer 3rd Edition * Data Structures with C, Schaum’s Series | | |
| **Grading Policy:** | * 40% Mid Tem Exam+ Sessional (Assignments, Quizzes, Project) * 60% Final Exam Theory | | |
| **Plagiarism Policy:** | Collaboration and group work is encouraged but each student is required to submit his/her own contribution(s). Your writings must be your own thoughts. Cheating and plagiarism will not be tolerated and will be referred to the Dean for appropriate action(s). | | |
| **Quizzes /Assignments Policy** | **Quizzes:**   * Quizzes will be unannounced some time, but tentative plan of quizzes is given in the outline. * They will be taken either in the first ten minutes of the class (so come to the class on time & be prepared!) or in the last ten minutes of the class (so listen to the lecture carefully). * If you miss a quiz, you miss it! * It’s up to the instructor’s discretion to choose the number of quizzes for evaluation purposes.   **Assignments:**   * Tentative number of Assignments is given in the outlines. * Assignments will be submitted on time. No late submission will be allowed.   Copying/Cheating whole or part of the assignment from anywhere without proper credit/references will not be tolerated. Whether you have copied or your work has been copied by someone else, you will get zero marks in that assignment. | | |
| **Project Work** | Students will be given a practical nature problem and at the end, each group will make a presentation of his/her work and/or give a demo of the project. | | |

|  |  |  |
| --- | --- | --- |
| **Week** | **Description** | **Comments** |
| 1 | Basics of C++, Data Structures and Algorithms, Major Operations of Data structures |  |
| 2 | Algorithms, Types of Data Structures, Mathematical Background, What to Analyze, Running Time Calculations |  |
| 3 | Handling Arrays, Row Major and Column major Operations, Various operations on arrays, Matrix Handling and various operations  Algorithms, Examples and their implementation  String processing Algorithms |  |
| 4 | The QUEUE ADT, Operations, Algorithms, Examples and their Implementation  DEQUE, CIRCULAR QUEUE, Examples and their Implementations |  |
| 5 | The STACK ADT (Push, Pop, Use of Open Stack, Expression Evaluation, Conversion between Infix, Prefix, Postfix using various algorithms  Algorithms, Examples and their implementation ) |  |
| 6 | The STACK ADT (Push, Pop, Use of Open Stack, Expression Evaluation, Conversion between Infix, Prefix, Postfix using various algorithms  Algorithms, Examples and their implementation ) |  |
| 7 | The Lists ADT (Singly Link List, Doubly Link, Circular Link List; Algorithms, Examples and their implementation) |  |
| 8 | The Lists ADT (Singly Link List, Doubly Link, Circular Link List; Algorithms, Examples and their implementation) |  |
| 9 | **MID TERM** |  |
| 10 | Trees (Simple Tree, Binary Tree, BST, Red Black Tree, AVL Tree, Tree Traversal, Successor, Predecessor, Insertion Deletion)  Algorithms, Examples and their implementation |  |
| 11 | Trees (Simple Tree, Binary Tree, BST, AVL Tree, Tree Traversal, Successor, Predecessor, Insertion Deletion)  Algorithms, Examples and their implementation |  |
| 12 | Hashing, Hash Functions, Separate Chaining, Handling Hash tables  Algorithms, Examples and their implementation |  |
| 13 | Sorting(Bubble sort, Insertion Sort, selection sort, Merge sort, Quick Sort, Examples and their implementation |  |
| 14 | Graphs (Introduction, Representation, Traversal, Shortest-Path Algorithms, Examples and their implementation |  |
| 15 | Graph Algorithms(Graph Traversal Algorithms, Minimal spanning tree, Prims Algorithm, Kruskal Algorithm etc) Algorithms, Dijskstra’s algorithm, Examples and their implementation |  |
| 16 | Basics of Recursion, Function call and Recursion Implementation, Indirect Recursion, Nested Recursion, Examples and their implementation |  |
| **FINAL EXAML** | | |