**CENTRAL UNIVERSITY**

**SCHOOL OF APPLIED SCIENCES**

**DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY**

**Course Code**: **ITEC 205** **Credit Hour(s)**:**3** **Webpage** :www.mlearn.central.edu.gh

**Course Title**: Algorithm & Data Structure **Academic Year**: Sept 2022/23

**Course Lecturer: Joel K. Appiah Room: Block B**

**Email Address: jkappiah@central.edu.gh Tel no.: 0242779942**

**Office Hours:**

# Course Objective

This course introduces basic data structures and their application using the C++ programming language. We introduce a mathematical framework for evaluating the efficiency of algorithms, and develop implementations of basic data structures such as lists, stacks and queues. We study searching and sorting algorithms and introduce recursion as a strategy for improving the running time of these algorithms. This leads us to study more advanced data structures that are defined recursively, such as trees and heaps. We cover several advanced topics, such as hash tables and the storage and exploration of graphs.

# Course Description

The objective of the course is to introduce students to algorithm and data structures.

This course has two parts. The first part is an introduction to the design and analysis of efficient algorithms. It explores various techniques used to organize data in computer memory for easy access, modification and deletion. The second part focuses on building dynamic data structures. Topics include models of computation, NP-completeness, efficient sorting and searching and algorithms for algebraic problems. Others will delve into building dynamic data structures like linked lists, stacks, queues and trees. Preferably C++ will be used.

# Learning Outcomes

At the end of this course, you should have learned:

* How think about data and operations on data
* How to design data structure for efficient use
* How to determine the efficiency of an algorithm
* Basic data structures and algorithms
* More complex programming techniques

# Instructional Methods

Instructional approaches to be used during the course (e.g., lectures, seminars, laboratory activities, tutorials, group projects). Note that attendance is also a requirement.

# Required Course Materials and Readings

* Weiss, Mark A. *Data Structures and Algorithm Analysis in C++.* 4th Edition. Pearson 2014. ISBN‐13:978‐0‐13‐284737‐7
* Malik, D S. *Data Structures in C++*,(2nd Edition), Cengage Learning
* Chang, S. (2003). *Data structures and algorithms.* Hackensack, NJ: World Scientific.

Cormen, T.H., Leiserson, C. E., Rivest R. L. & Stein, C. (2009) *Introduction to algorithms* (3rd ed.). Cambridge, MA: MIT Press.

* Drozdek, A. (2000). *Data structures and algorithm in C++* (2nd ed.). St. Paul, MI: Brooks/ Cole.
* Goodrich, M. T., & Tanmassia, R. (2005). *Data structures and algorithms in Java* (4th ed.). Hoboken, NJ: Wiley.
* Preiss, B. R. (1999). *Data structures and algorithms with object-oriented design patterns in Java.* Michigan, MI: John Wiley.

# Evaluation

Class test, quizzes, mid-semester exams, term project and end of semester exams.

# Commit To Academic Integrity

Students in the department are expected to maintain **high degrees of professionalism, commitment to active learning, participation and academic integrity every time**.

# Academic Dishonesty

Please note that students involved in academic dishonesty will receive a **ZERO** mark on the particular component in which the infraction occurred and a notation of academic dishonesty in the departmental office. This may also reflect on references written by the department.

**It is the student’s responsibility to understand what constitutes academic dishonesty.**

# Missed Exams / Tests / Assignments

**Assignment Submission**: Assignments must be received on the due date specified for the assignment.

**Lateness Penalty:** Assignments received later than the due date will be penalized Exceptions to the lateness penalty for valid reasons such as illness, etc., may be entertained by the Lecturer but will require supporting documentation (e.g., a doctor’s letter).

**Missed Tests:** Students with a documented reason for missing a course test, such as illness, which is confirmed by supporting documentation (e.g., doctor’s letter) will be handled by the Lecturer.

**WEEK BY WEEK COURSE SCHEDULE / ORGANISER:**

|  |  |  |
| --- | --- | --- |
| Week | Topic |  |
| Week 1 | **Introduction**  Review: data structure concepts, arrays |  |
| Week2 | **Analysis of Algorithms**  What to analyze. Analysis techniques. Efficiency of  algorithms. Comparing efficiency of various algorithms  Rates of Growth:  O(n),Ω(n),Θ(n),o(n),ω(n), Run-Time Complexity, Space Complexity, NP-Completeness (Time Permitting), etc | Assignment 1 |
| Week3 | **Algorithm Design Techniques**   * Greedy Algorithms * Divide and Conquer * Dynamic Programming * Randomized Algorithms * Backtracking Algorithms |  |
| Week4 | **Searching and Hashing algorithms**.  Search algorithms – Sequential Search, Ordered lists, binary search. Searching using  Hashing. Hash tables. Hash functions. Some examples of hash functions. Collision resolution. |  |
| Week5 | **Sorting Algorithms**  Insertion Sort, Selection Sort, Merge Sort, |  |
| Week6 | **Sorting Algorithms**  Quicksort, Heapsort, Bucket Sort, Radix Sort, etc | Assignment 2 |
| Week 7 | **Basic Data Structures**  Lists, Stacks  Behavior of a Stack. Basic operations on a Stack. Array-based stacks. Linked-list based implementation. Expression evaluation using a stack. |  |
| Week 8 | Queues and Heaps  Behavior of a queue. Basic queue  operations Study implementations using an array and a linked-list. | Mid Semester Exams |
| Week 9 | Linked-List. Singly linked-lists.  Implementation using pointers. Basic Operations |  |
| Week 10 | **Trees**  Binary and non- binary trees. Structure of a binary tree.  Definitions and properties. Traversing a binary tree. | Assignment 3 |
| Week 11 | **Binary Search Tree** (BST). Organizing data in a BST. Inserting and deleting items in a BST |  |
| Week 12 | **Graph Algorithms**  Definitions  Topological Sort  Shortest  -Path Algorithms  Network Flow Problems  Minimum Spanning Tree  Applications of Depth  -First Search  Introduction to NP-Completeness |  |
| Week 13 | **Graphs and Graph Algorithms**  Dijkstra’s Algorithm,  Depth-First vs Breadth-First  Search | Assignment 4 |
| Week 14 | Revision |  |
|  |  |  |