SYLLABUS

COURSE	CSCI 373 (Advanced Data Structures)
INSTRUCTOR	Maksi Kutrolli
E-MAIL	mkutrolli@jjay.cuny.edu
OFFICE	6.65.24
COURSE WEBPAGE	Blackboard

LEARNING OBJECTIVES:

At the end of this course, students will be able to:

1- Understand the concepts of Object-Oriented Programming.

2- Master the use of pointers in C++ and their role in memory management and data manipulation.

3- Understand the benefits of data structures. Define the ADT and Concrete implementation concepts. Learn about Complexity Analysis (Big O Notation) and its importance in analyzing algorithm performance.

4- Explore the intricacies of linked lists, including singly linked lists, doubly linked lists, and circular lists, and their applications in data storage and manipulation, implementing each type as practical examples.

5- Understand the concepts of stacks and queues, their basic operations, and real-world use cases in managing data, and implement these data structures.

6- Explore the use of lists and iterators, focusing on vectors, lists, and sequences as data structures, and understand how they contribute to efficient data management, implementing each data structure.

7- Delve into general trees and learn about various tree traversal algorithms used to navigate and manipulate tree structures, implementing different types of tree structures. Understand the structure and applications of binary trees, which are fundamental in computer science, and implement binary tree structures.

8- Learn about the Priority Queue Abstract Data Type (ADT) and its role in managing data with varying priorities, and implementing priority queues.

9- Explore the concepts of hash tables and maps and understand how they provide efficient data retrieval and storage, implementing these data structures.

10- Gain insight into dictionaries and graph algorithms, which are essential in solving complex problems in computer science, and implement dictionary structures and graph algorithms.

11- Implement and use suitable data structures for a given basic problem in computing.

DESCRIPTION:

This course explores advanced data structures in C++ with a focus on fundamental concepts including pointers, arrays, abstract data types, and complexity analysis. It covers a diverse range of structures such as linked lists, stacks, queues, vectors, and iterators. Students will delve into sophisticated topics like trees, priority queues, hash tables, maps, and graph algorithms, emphasizing practical applications. They will gain hands-on experience through projects implementing data structures from scratch using templates. Students will develop essential skills in data organization, algorithmic analysis, and C++ software development.

The prerequisites for this course are Eng 201 and CSCI 272 (MAT 272). Students are expected to have strong knowledge of and be able to program in C++ in order to be able to implement data structures.

TEXTBOOK:

Data Structures and Algorithms in C++ (2nd edition) by Michael T. Goodrich, Roberto Tamassia, David M. Mount
ISBN-13 978-0-470-38327-8
This is the main book that we will use in this course.

Reference (optional)
- Introduction to Algorithms by Thomas Cormen.
- A free book available at http://opendatastructures.org/ods-cpp.pdf
- Data Structures and Algorithms in C++ (4th edition) by Adam Drozdek
  ISBN-13: 978-1-133-60842-4

METHOD OF ASSESSMENT:

- 1 Midterm and Final exam ---50%
- Assignments ---35%
- Discussion and class participation ---15%

Note: Grading of assignments and exams can be reviewed in person by the professor to ensure that the submitted work is the student's own code and to address any questions regarding the understanding of the submitted material.

QUIZZES:

During the semester, students may be given quizzes from the topics covered during the previous lectures. This is done to encourage students to cultivate the habit of preparing for the classes weekly, and hence be ready to learn new concepts the following weeks. This also helps them to prepare for the examination. Generally, a quiz will be given at the beginning of a class and students who are absent or late for the class will not be allowed to take that quiz at a later time.
Quizzes will be counted toward assignment grades

PROGRAMMING PROJECTS:

This course emphasizes on the ability of the students to implement efficiently any given data structure, using a high-level programming language. Therefore, students are expected to complete several programming projects that require the implementation of several basic data structures, using the C++ programming language.

RESPONSIBILITIES:

Students are expected to attend the class lectures and take the exams at the scheduled times. Assigned readings, exercises and projects must be completed on time.

ACADEMIC INTEGRITY

Cheating on exams or copying projects will not be tolerated. Please review the College's policies on Plagiarism and Cheating. Moreover, if you copy projects or assignments, you will not be able to answer questions on the exams.

IMPORTANT WEBSITES

- Data Structures and Algorithms in C++ publisher's website This is a link to the publisher's website where you would a have a link to download data files and source code.
- CUNY Blackboard: All course related assignments and material will be available on

blackboard. Announcements will be posted on blackboard. Students must have a working CUNY portal account in order to receive course related information

PROPOSED SCHEDULE OF LECTURES:

Week	Topics	Chapters or sections	Assignments
1	**Review of Object-Oriented Programming Using C++** (Classes & Objects, Encapsulation, Friendship, Inheritance, Polymorphism, Abstract Classes, Interface)	*1.5, 2.1, 2.2*	**Assignment 1 Assignment 2 Due in 10 days**
2	**Pointers & Arrays**	1.1.3, 3.1	
3	**Data Structure** (Abstract Data Type, Concrete Implementation) **Complexity Analysis** (Big O Notation)	2.1, 2.2.5, 4.2	**Assignment 3 Due in 7 days**
4	**Linked List** (Singly Linked Lists, Doubly Linked Lists, Circular Lists)	3.2, 3.3, 3.4	
5	**Recursion**	3.5	**Assignment 4 Due in 7 days**
6	**Stacks and Queues**	5.1, 5.2	
7	**Midterm Exam**	Week 1 to 6	**Quiz and midterm project**
8	**List and Iterator** (Vectors, Lists, Sequences)	6	
9	**General Trees Tree Traversal Algorithms**	7.1, 7.2	**Assignment 5 Due in 7 days**
10	**Binary Trees**	7.3	
11	**Priority Queue ADT**	8.1, 8.2	**Assignment 6 Due in 7 days**
12	**Hash Tables, Maps**	9.1.1 to 9.1.3, 9.2.1 to 9.2.4, 9.2.5 to 9.2.7	
13	**Dictionaries and Graph Algorithms**	9.5.1, 9.5.2, 13.1, 13.2	**Assignment 7 Due in 7 days**
14	**Review**		
15	**Final Exam**		**Quiz and final project**