NEW YORK CITY COLLEGE OF TECHNOLOGY/CUNY
Computer Systems Technology Department

COURSE: CST3650 – Data Structures (4 hours , 3 credits)
INSTRUCTOR:
E-MAIL:
OFFICE HOURS:
Extension

If you do not do the homework programs, you cannot pass the course. If home work is late, 1 point (25%) is deducted. After one week of lateness, home work will not be accepted.

Course Description:

A data structure is an arrangement of data in a computer's memory. In this course you will learn ways to structure and arrange data in computer's memory that includes **array, linked lists, stacks, trees, hash tables, graphs**, **among others**. Algorithms manipulate that data in these structures in various ways, such as searching for a data item and sorting a set of data elements. In this class you will learn several concepts including how to organize data elements, how to delete, insert, edit and search for a data element in a specific data structure and how to sort a set of data elements. You will also learn the differences between different data structures and when to use to use the right ones to solve problems.
Motivation.
Students will understand the importance of the emerging field of data science and its intersection with data structures in computer science. Students will acquire more skills in computer programming, helping them in the job market.

Course Objectives:

Upon completion of this course, student will be able to:

1. Demonstrate some understanding of object-oriented programming and be able to program with classes.
2. Demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, and trees and be able to use these structures effectively in application programs.
3. Implement various data structures in more than one manner, compare the different implementations and explain the advantages and disadvantages of the different implementations.
4. Demonstrate understanding of and be able to program various sorting algorithms, and be able to compare the efficiency of these algorithms in terms of both time and space.
5. Trace and code recursive functions.

6. Demonstrate understanding of and be able to represent graphs, explore graphs as ADT, examine and implement various graph traversal algorithms, the shortest path algorithm, and the minimal spanning tree algorithm.

General Education Outcomes:

- **SKILLS/Inquiry/Analysis:** Students will employ scientific reasoning and logical thinking.
- **SKILLS/Communication:** Students will communicate in diverse settings and groups, using written (both reading and writing), oral (both speaking and listening), and visual means
- **INTEGRATION/Integrate learning:** Students will resolve difficult issues creatively by employing multiple systems and tools.

- **VALUES, ETHICS, RELATIONSHIPS / Professional/Personal Development:** Students will work with teams, including those of diverse composition. Build consensus. Respect and use creativity.

Prerequisites:

CST3503 or CST3513

Required Textbook:

Data Structures and Algorithms in C++, 4th Ed, by Adam Drozdek, Cengage learning 2013.

Academic Integrity Policy - Students and all others who work with information, ideas, texts, images, music, inventions, and other intellectual property owe their audience and sources accuracy and honesty in using, crediting, and citing sources. As a community of intellectual and professional workers, the College recognizes its responsibility for providing instruction in information literacy and academic integrity, offering models of good practice, and responding vigilantly and appropriately to infractions of academic integrity. Accordingly, academic dishonesty is prohibited in The City University of New York and at New York City College of Technology and is punishable by penalties, including failing grades, suspension, and expulsion. The complete text of the College policy on Academic Integrity may be found in the catalog.

Suggestions:

You are encouraged to solve all the homework questions and programs on your own, but are permitted to brainstorm difficult problems in small groups, as long as each of you writes the solutions individually and honestly acknowledging the cooperation. Needless to say, if you work with others but never come up with the solution on your own, you may do OK in the homework component of your grade, but you will not do well on exams, so be careful. The use of internet, course bibles, outsiders, and other clearly "cheating" resources is strictly prohibited. Students found cheating will fail the course. If you miss a class, you are responsible for the material. Have the phone numbers of several other students to find out what material you missed

Grading Procedure:

Final Examination	35%
Midterm Examination	30%

Tests	20%
Assignments	15%
Total	100%

Course Outline

	Topics	Reading
Week 1,2	Review of Object-Oriented Programming Object-Oriented Design	Chapter 1
Week 3	Complexity Analysis	Chapter 2
Week 4	Stacks and Queues	Chapter 4
Week 5	Review and Test1	
Week 6	Recursion	Chapter 5
Week 7,8	Linked Lists Review and Test2	Chapter 3
Week 9	Binary Search Trees	Chapter 6
Week 10	Sorting	Chapter 9
Week 11,12	Hashing Review and Test3	Chapter 10
Week 13,14	Graphs	Chapter 8
Week 15	Review	
Week 16	Final Exam	

Assessment criteria

For the successful completion of this course a student should be able to:	**Evaluation methods and criteria**
1. Understand how to use arrays, Structures	1. Students will write programs that use: • Arrays-one-dimensional and two-dimensional • Records or Structures • Classes, Inheritance, Polymorphism, Overloading, etc..
2. Understand the use of tools to analyze an algorithm.	2. Students will write programs to prove the correctness and speed of algorithms.
3.Understand stacks , Queues	3. Students will write programs to search insert and delete an element from stacks and queues. Apply stacks to evaluate postfix and prefix expressions
4. Understand Recursion	4. Students will write programs to implement recursion principles such as factorial, Tower of Hanoi, Fibonacci sequence etc..
5. Understand Linked Lists	5. Students will write programs to search insert and delete an element from a single and double linked list. Programs will be written to implement stacks and queues as linked list.
6. Understand the use of tree as data structure	6. Students will write algorithms for different types of traversals. Finding, inserting and searching for a node in a binary search tree. Students will write programs to implement AVL trees, Multi-Way search trees, etc..
7. Understand how to search and sort in a list of data, and hashing	7. Students will write algorithms to use linear search, binary search, Quick-Sort, Insertion-Sort, Merge-Sort, Heap-Sort, and hash functions
8. Understand Graphs a data structure	8. Students will write programs to do Graph traversals: depth first search and breath first search. Graph connectivity. Algorithms to implement Shortest paths, Minimum Spanning Trees, etc..

General Education Outcomes and Assessment:

Learning Outcomes	**Assessment Method**
SKILLS/Inquiry/Analysis Students will employ scientific reasoning and logical thinking.	Students will describe problem, identify inputs, processes and desired outcomes in class work and tests. Students will find algorithm to solve problems in class work and tests. Students will identify coding paradigms in class work and tests
SKILLS/Communication Students will communicate in diverse settings and groups, using written (both reading and writing), oral (both speaking and listening), and visual means	Term paper in which students will describe the programming project and how it was implemented using an object-oriented programming Language, and the right data structure Oral presentations of the programming project
INTEGRATION/Integrate learning: Students will resolve difficult issues creatively by employing multiple systems and tools.	Programming assignments where student will write codes to solve problems independently.
VALUES, ETHICS, RELATIONSHIPS / Professional/Personal Development Students will work with teams, including those of diverse composition. Build consensus. Respect and use creativity.	Programming project in which students will work in groups, build consensus and respect and use creativity

Bibliography

1. Data Structures Using C++, By D.S. Malik, ISBN 0-619-15907-3, Thomson, Course Technology
2. Data Structures and Algorithm Analysis in JAVA, 3rd Ed, By Mark Allen Weiss, ISBN10 0-13-257627-9, Pearson
3. Data Structures Using C and C++, 2nd Ed, By Langsam, Augenstein, and Tenenbaum, ISBN 0-13-036997-7, Prentice Hall
4. Data Structures with C++ Using STL, 2nd Ed, By Ford and Topp, ISBN 0-13-085850-1, Prentice Hall
5. Object-Oriented Data Structures Using Java, 4th ed, By Nell Dale, Daniel T. Joyce, Chip Weems ISBN-13: 9781449613549, Jones & Bartlett Learning