HUDSON COUNTY COMMUNITY COLLEGE
COURSE SYLLABUS

Course Name: CSC 214 - Data Structures and Advanced Programming
Credit: 3
Prerequisite: CSC 115 or CSC 117
Location:
Instructor:
Phone:
E-mail:

Course Description

This course examines data structures and their software implementation. The topics covered include top-down design; Topics considered include Pointers lists, strings, stacks, queues, trees, graphs, networks, file structures, recursive functions, sorting techniques, hashing, and the analysis of algorithms.

Data Structures and Algorithms is a study of the representation and Implementation of abstract data types and related algorithms that are used in computer science

Course Objectives

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

- To design a program using structured development techniques.
- Implement structured design using functions.
- Understand the structure and processing of single and multi-dimensional arrays.
- Organize and represent data using structures.
- Understand the fundamental techniques for sorting and searching data sets. and to identify the advantages and disadvantages of each.
- Understand the concept of an abstract data type.
- Know the fundamental order of magnitude growth rates and how they are used to measure the run-time efficiency of an algorithm.
- Understand the principles of pointers, dynamic memory management, and be able to apply these concepts in constructing dynamic data structures.
- Know the fundamental properties of stacks and queues and be able to implement them using dynamic linked-lists.
- Incorporate recursive techniques in the representation and implementation of an abstract data type.
- Understand the fundamental properties of binary trees, binary search trees, and general trees and be able to implement them using dynamic data structures.
- Represent graphs and networks using adjacency matrices and adjacency lists and implement them using the appropriate data structure.

EVALUATION CRITERIA:

Student will be graded based on:

Midterm	30%
Programming Projects	20%
Final	30%
Homework, group work and quizzes	20%

Grading Policy:

90 to 100	A
85 to 89	B+
80 to 84	B
75 to 79	C+
70 to 74	C
60 to 69	D
00 to 59	F

There will be no makeup for missing tests unless official documents are presented.

Late Homework assignments:

Programming assignments:

a) ***Up to 3 days late, 30% penalty.***
b) ***Beyond 3 days, not accepted.***

None programming assignments:

Not accepted after due date and time

Any student misses a class for any reason is responsible for the notes and the assignments that are given on the day he/she missed.

The schedule for the tests and the laboratory assignments depends on the covered material.

Excess of absence will result in a failing grade (6 absences maximum).

**** 20 minutes of lateness is considered one absence.**

**** failure to keep Camera on or not showing your face during class time is considered absence (unless pre authorized by the instructor)**

. If students are unable to turn their cameras on, they should communicate the circumstances to the faculty member. On-campus spaces are also available to students as an alternative to home or off campus online and remote instruction. The on-campus spaces include: Gabert Library L219, L221, L222, L419, STEM Building S217, and North Hudson Campus N224, N303D. Within these rooms, students will have access to computers, web cameras, and headsets. If there are any issues with space capacity, there are several additional rooms that can be utilized.

Text book:

1. **Starting out with C++** (Gaddis 8th or 9th edition)
2. Instructor provided PPTs and handouts

References:

1. Data Structures, A pseudocode approach with C++, by Richard F. Gilberg and Behrouz AForouzan ISBN: 0534-95216
2. Algorithms and Data Structures in C++, Allan Parker, ISBN 0-3493-7171-6
3. Data Structures and Algorithms in Java, Peter Drake, ISBN 0-13-346-914-2
4. Java Software Structures, 4th edition, by Lewis and Chase, ISBN 13: 978-0-13-325012-1
5. Starting out with Java, from Control structures through data structures, ISBN: 9780134787961, 013478796X

Programming Projects

Programming projects should be entirely your own work. The following are prohibited:

- Copying portions of a programming code from another student (or from any other source, including your textbook, other books, or the Internet) into a programming code for your project
- Allowing another student to copy your programming code

If the above rules are violated, you will receive an "F" for your project
It is acceptable to talk in general terms about the project without getting into details about how to actually program it. In addition, you may seek assistance from tutors, both in and out of class, however; the program that you submit **must** be your own.

ACADEMIC INTEGRITY

Academic integrity is central to the pursuit of education. For students at HCCC, this means maintaining the highest ethical standards in completing their academic work. In doing so, students *earn* college credits by their honest efforts. When they are awarded a certificate or degree, they have attained a goal representing genuine achievement and can reflect with pride on their accomplishment. This is what gives college education its essential value.

Violations of the principle of academic integrity include:

- Cheating on exams.
- Reporting false research data or experimental results.
- Allowing other students to copy one's work to submit to instructors.
- Communicating the contents of an exam to other students who will be taking the same test.
- Submitting the same project in more than one course, without discussing this first with instructors.
- Submitting *plagiarized* work. *Plagiarism* is the use of another writer's words or ideas without properly crediting that person. This unacknowledged use may be from published books or articles, the Internet, or another student's work.

When students act dishonestly in meeting their course requirements, they lower the value of education for all students. Students who violate the college's policy on academic integrity are subject to failing grades on exams or projects, or for the entire course. In some cases, serious or repeated instances of academic integrity violations may warrant further disciplinary action.

Disability Support Services

Students with disabilities who believe that they might need accommodations in this class are encouraged to contact the Disability Support Services at 201-360-4157 as soon as possible to better ensure that such assistance can be implemented in a timely fashion. All disabilities must be documented by a qualified professional such as a physician, licensed learning disability teacher (LDTC), psychologist, psychiatric nurse, licensed social worker or licensed professional counselor, who is qualified to assess the disability that the student claims to have and note recommendations on accommodations for the student. All information provided to the Disability Support Services Program will be confidential between the program, professors involved with the student, and the individual student.

Student Classroom Recording Policy

- Hudson County Community College prohibits the audio-visual recording, transmission, and distribution of classroom sessions. Classes may only be recorded with the advance written permission of the instructor. The Hudson County Community College classroom recording policy must be listed in all syllabi.
- All classroom recordings can only be used for academic purposes by students enrolled in that class. Recordings may not be shared, reproduced, or uploaded to public websites or other mediums, and these recordings may contain copyrighted material and are prohibited from any form of commercial use.
- All students and guests must be informed that the class may be recorded. Due to issues related to privacy and the possible inhibition of student participation, instructors should be mindful of the effects of permitting classroom recording.
- Instructors should retain electronic or paper copies of their written consent to grant classroom recordings.
- Students must destroy their recordings at the end of the semester.
- Students who are granted permission to record their class by the office of Disability Support Services should inform the instructor beforehand and are subject to the policies outlined in this document.
- Violation of this policy is subject to disciplinary action listed under the code of conduct as included in the Student Handbook.

	Topic	Lab	Homework
1	Syl, and introduction		
	Abstract Data Types Abstract data types (ADTs) in program design Data structure selection criteria		HW 14
2	Analysis of Algorithms Worst and average case behavior Order of magnitude analysis of algorithms Expected run-time of an algorithm	programming analyses	HW 19
3	Arrays and Functions (Methods) Arrays as function arguments Array processing algorithms Functions Function declaration, definition, and call Return statement Reference types and pass by reference Menu-driven systems	Programming lab and Example	 Programming HW
5	Programming Arrays/functions Sequential Access Files Record and file terminology Structures and nested structures Binary file organization Creating and processing a sequential access		Programming assignment

6	Searching (Sequential Search and Binary Search) Hashing (techniques and collision resolution	Hashing assignment 86, Searching 87
7	Recursion Recursive functions and algorithms Iterative versus recursive implementations	Recursion Handout
8	**Review and Midterm Exam**	
9	Sorting Techniques Criteria for selecting a sorting algorithm Quick sort ADT: heap and heap sort algorithm Radix sort algorithm Shell sort algorithm	sorting program
10	External sorting – merge sort algorithm (Merge, Quick, Heap)	Sorting program
11	Stacks concept Terminology and fundamental operations Infix, prefix, and postfix notation Algorithm to convert infix form to postfix form Parsing and evaluating arithmetic expressions	Stacks Implementation (Arrays)
12	Queues Concept ADTs: Queue, Priority Queue, and Dequeue Terminology and fundamental operations Circular implementation of a queue Priority queue: analysis and implementation	Queue Implementation (Arrays)
13	Pointers, Lists, Linked List and Stacks Dynamic linked-list implementation of lists Load list and dump list algorithms Insertion and deletion of list elements Other list processing algorithms	Array implementation versus dynamic linked-list implementation LL and stacks handout
14	Introduction to Trees ADTs: Binary Tree, Binary Search Tree, and General Trees Tree terminology Trees Traversal Graphs and Shortest path Alg (Dijkestra) Graph terminology and graph traversals	Trees handout Fundamental algorithms for graphs and networks ADTs: Graph and Network Adjacency matrix representation Adjacency list representation
15	Review and Final Exam	