

Common Course Outline

CSIT 211

Advanced Programming

4 Semester Hours

The Community College of Baltimore County

Description

CSIT 211 – 4 credits – Advanced Programming teaches skills for solving complex problems; discusses advanced data structures and algorithms for recursion, pointer variables, linked lists, stacks, queues, hash tables, collections, sorting and searching, and trees.

4 credits: 4 lecture hours per week

Prerequisite: A letter grade of B or better in CSIT 210 (same as CINS 236 or CMSC 201), or consent of Program Director.

Overall Course Objectives

Upon successfully completing the course, students will be able to:

1. analyze problems to determine most appropriate data structure;
2. use an object-oriented or structured programming language for complex problem solving;
3. develop well-written and documented programs that use recursion, dynamic memory allocation, and abstract data types;
4. evaluate mathematical efficiency of algorithms and sorting and searching techniques;
5. explain the advantages and disadvantages of object-oriented languages compared to procedural programming languages;
6. apply abstract data types to applications;
7. use recursion to solve looping problems;
8. explain advantages of encapsulation and inheritance; and
9. work in teams to develop large complex programs.

Major Topics

I. Introduction to Abstract Data Types

- a. Definition of ADT
- b. Use of Specification and Implementation Files
- c. Collections

II. Recursion

- a. Static versus Dynamic Storage Allocation
- b. Recursive versus Iterative Solutions

III. Databases

- a. Basic table and data commands
- b. Connecting to a Database

IV. Abstract Data Types - Implementations and Applications

- a. Linked Lists

- b. Stacks
 - c. Queues
 - d. Binary Search Trees
 - e. Various Tree Traversals
 - f. Heaps
 - g. Collections
- V. Big-O Notation and Efficiency of Algorithms
- a. Definition
 - b. Examples
- VI. Sorting Algorithms
- a. $O(N^2)$ algorithms
 - b. $O(N \log N)$ algorithms
- VII. Searching Algorithms
- a. Sequential Search
 - b. Binary Search
 - c. Hashing
- VIII. Object-Oriented Programming Topics
- a. Encapsulation
 - b. Inheritance
 - c. Polymorphism and Dynamic Binding
 - d. Operations in Derived Classes
 - e. Constructors in Derived Classes
- IX. Advanced Sorting/Searching Techniques (optional)
- a. Binary Sort
 - b. Shell Sort
 - c. 2-3 Trees
- X. Advanced Graphical User Interfaces
- a. Timers
 - b. Mouse Interface features

Course Requirements

Grading: Grading procedures will be determined by the faculty member, will be provided the first week of class, and will include:

1. Minimum of 5 programming projects*
2. Minimum of 2 tests
3. Comprehensive final exam or programming project

*These projects will include collaborative work, written portions and oral presentations as assigned by the faculty member.

Other Course Information

This course is the second of a sequence of two transfer courses in the Computer Science Program at CCBC. Many 4-year colleges recommend that Computer Science majors complete both courses in the sequence for transferability.

CSIT 211 is required for the CGVC – SDE (Simulation and Digital Entertainment) option.