

# Common Course Outline

CMSC 202

Computer Science II

4 Semester Hours

## The Community College of Baltimore County

### Description

#### **Computer Science II for Majors**

Teaches skills for solving complex problems; discusses advanced data structures and algorithms for recursion, pointer variables, linked lists, stacks, queues, hash tables, and trees.

Prerequisite: CINS 225 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.
9. work in teams to develop large complex programs.
10. continue the study of advanced computer science at a 4-year college.

### Major Topics

#### I. Introduction to Abstract Data Types

- A. Definition of ADT
- B. Use of Specification and Implementation Files

#### II. Recursion

- A. Static versus Dynamic Storage Allocation
- B. Recursive versus Iterative Solutions

#### III. Use of Pointers and Dynamic Memory Allocation

- A. Point variables
- B. Allocating and De-allocating dynamic memory
- C. Recursion using pointer parameters

#### 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. Graphs (optional)
- V. Big-O Notation and Efficiency of Algorithms
  - A. Definition
  - B. Examples
- VI. Sorting Algorithms
  - A.  $O(N^2)$  algorithms
  - B.  $O(N \cdot \log N)$  algorithms
- VII. Searching Algorithms
  - A. Sequential Search
  - B. Binary Search
  - C. Hashing
- VIII. Object-Oriented Programming Topics (Optional)
  - 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

## **Course Requirements**

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

1. At least five computer projects: Students will develop several computer programming projects, ranging from introductory labs to complex multi-layered scientific or mathematical projects. Programming time outside of classes is required to complete projects.
2. At least two Tests, Exams, and/or Quizzes: Individual faculty will notify students of the testing procedures to be used.
3. Comprehensive Final Exam: The course will include a comprehensive final exam, which may include a final project.
4. Final Grades: Grades will be determined by individual faculty members.

Individual faculty members may include additional course objectives, major topics, and other course requirements to the minimum expectations stated in the Common Course Outline.

The Community College of Baltimore County is committed to providing a high-quality learning experience that results in growth in knowledge, attitudes, and skills necessary to function successfully as a transfer student, in a career and as a citizen. To accomplish this goal, we maintain high academic standards and expect students to accept responsibility for their individual growth by attending classes, completing all homework and other assignments, participating in class activities and preparing for tests.

We take seriously our responsibility to maintain high-quality programs and will periodically ask you to participate in assessment activities to determine whether our students are attaining the knowledge, attitudes and skills appropriate to various courses and programs. The assessment activities may take many different forms such as surveys, standardized or faculty-developed tests, discussion groups or portfolio evaluations. We ask that you take these activities seriously so that we can obtain valid data to use for the continuous improvements of CCBC's course and programs.

### **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.