

**Common Course Outline**  
**CMSC 220**  
**Introduction to Assembly Language Programming**  
**3 Semester Hours**

**The Community College of Baltimore County**

**Description**

**Introduction to Assembly Language Programming**

Teaches number systems and radix conversion; discusses binary, hexadecimal representations of data, computer organization and addressing techniques, and assembly language instruction.

Prerequisite: CMSC 201 or CMSC 225 or CINS 225 or CMSC 107 or consent of instructor.

**Overall Course Objectives**

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

1. describe assembly language in relation to computer architecture
2. convert numbering systems
3. use internal registers
4. operate the stack
5. program with memory addressing techniques
6. debug programming errors with DOS debugger
7. develop loops and jumps
8. use multiple file-linking techniques
9. convert input from the user files
10. create arrays
11. develop procedures and macros
12. work in teams to develop a large program

**Major Topics**

- I. Basics of Numbering Systems
- II. Hardware
  - A. CPU and Internal Registers
  - B. Memory Locations
- III. Assembly Language Structure
  - A. Segments
  - B. Comments
- IV. Input/Output
  - A. DOS ROM BIOS Calls
  - B. Numeric I/O
  - C. Keyboard Input
- V. Data Movement
  - A. Instructions for Data Movement, byte swapping
  - B. Operation and use of stack
- VI. DOS Debugger
  - A. Program stepping, breakpoints

- B. Deassembly, memory and register examination and modification
- C. Debugging techniques
- VII. Arithmetic
  - A. Negative numbers, two's complement arithmetic, sign magnitude number representation, one's complement
  - B. Addition, subtraction, multiplication, division
- VIII. Comparing and Branching
  - A. Decision making
  - B. Loops
  - C. Arithmetic and conditional jumps
  - D. Unsigned conditional jumps
  - E. Unconditional jumps
- IX. Procedures
  - A. Call and return instructions
  - B. Use of stack for subprograms
  - C. Procedures to separate files
  - D. Multiple file-linking techniques
- X. Macros
  - A. Introduction
  - B. Pseudo-macros
- XI. Logical Operations
  - A. Boolean operations
  - B. Bit operations
  - C. Shift operations
- XII. Memory Addressing Techniques
  - A. Arrays
  - B. Indexed addressing
  - C. Base-indexed addressing
  - D. Look-up tables
- XIII. Segments
  - E. Offsets
  - F. Segment registers
  - G. Program segment prefix

## **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. Computer Projects: Students will develop at least five computer programming projects, ranging from introductory labs to complex multi-layered scientific or mathematical projects. Programming time outside of class 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 a transfer course in the Computer Science Program at CCBC.