CSIT 111
Fundamentals of Logic and Design
3 Credits

Community College of Baltimore County
Common Course Outline

Description

CSIT 111 – Fundamentals of Logic and Design: provides an introduction to programming concepts, logic, and design used in software development. This course emphasizes problem solving through the design of algorithms using techniques such as flow charting, pseudocode, and functional decomposition. Students implement algorithms using a current programming language. Topics include simple data types, arrays, functions, file input/output (I/O), control structures, and object-oriented development concepts.

Pre-requisites: ACLT 053 or (ESOL 052 and ESOL 054) and MATH 082

Overall Course Objectives

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

1. define the steps of the software development life cycle;
2. analyze problems through decomposition and abstraction;
3. design algorithms that will be translated into working solutions;
4. implement algorithms using a current programming language;
5. demonstrate testing, debugging, and validating solutions;
6. demonstrate the use of variables with valid data types;
7. design programs that perform accurate calculations to solve problems;
8. distinguish between control structures, sequence, selection, and repetition;
9. construct programs using control structures and functions;
10. demonstrate how to input and output data from the keyboard and files;
11. develop algorithms using arrays, arithmetic operators, and Boolean expressions;
12. discuss ethical issues surrounding secure programming, privacy, and intellectual property, and the implications of these topics in the context of software development;
13. analyze how local and global business impacts software development;
14. find, evaluate, use, and cite academic resources to research and communicate information needed for software development; and
15. demonstrate effective oral, written, and critical thinking skills as they relate to logic and programming design.

Major Topics

I. Overview of logic and programming languages
II. Algorithm design tools and techniques
III. Software development life cycle
IV. Decomposition and abstraction
V. Variables and data types

VI. Boolean and arithmetic expressions
VII. Input and output
VIII. Selection Statements
IX. Repetition Statements
X. Functions
XI. File I/O
XII. Arrays
XIII. Debugging techniques
XIV. Object-oriented concepts
XV. Discipline-specific ethical issues

Course Requirements

Grading will be determined by the individual faculty member, but shall include the following, at minimum:

- Five quizzes
- Five programming projects
- Midterm exam
- Final exam
- One term project

Written assignments and research projects: Students are required to use appropriate academic resources in their research and cite sources according to the style selected by their professor.

Other Course Information

This course is an approved 3–credit General Education course in Information Technology. One or more assignments will infuse CCBC General Education Program outcomes and will account for a minimum of 10% of the total course grade. The assignment(s) will allow students to demonstrate at least 5 of the 7 General Education program outcomes.

Date Revised: 9/6/2022