IT-1050: PROGRAMMING LOGIC

Cuyahoga Community College

Viewing: IT-1050 : Programming Logic

Board of Trustees:
January 2024

Academic Term:
Fall 2024

Subject Code
IT - Information Technology

Course Number:
1050

Title:
Programming Logic

Catalog Description:
Learn to solve business problems by designing, coding, and testing programming solutions using a current high-level programming language. Learn and apply standard language constructs, control flow, and beginning object-oriented programming concepts.

Credit Hour(s):
3

Lecture Hour(s):
2

Lab Hour(s):
2

Requisites

Prerequisite and Corequisite
IT-1025 Information Technology Concepts for Programmers, or concurrent enrollment; and
MATH-0955 Beginning Algebra; or co-enrollment in a co-requisite pairing of MATH-0930 Essential Skills for Algebraic & Quantitative Reasoning and MATH-1190 Algebraic & Quantitative Reasoning; or qualified Math placement.

Outcomes

Course Outcome(s):
Analyze, design and test programs to address specified business problems utilizing programming logic including object-oriented and structured concepts.

Objective(s):
1. Analyze and research simple business problems in order to design effective algorithms and programming solutions.
2. Identify, analyze, and research given programming problems to define necessary inputs, outputs, and processes.
3. Write code with control flow statements, including decisions and loops, to change the order statements in a program are executed.
4. Explain the design and advantages of modularization.
5. Create and call methods with the appropriate access modifier, return type, naming, parameters, arguments, and implementation.
6. Demonstrate the use of a simple data structure.
7. Explain the design and demonstrate the use of classes and objects.
8. Explain object-oriented design concepts such as inheritance, encapsulation, and polymorphism.
9. Write and perform tests to confirm code validity.

Course Outcome(s):
Perform as both a team member and individually, in a professional environment where the business and technical environment are constantly changing.

Objective(s):

1. Engage in directed work as a member of a software development team.
2. Perform skills as a self-starter, demonstrating the ability to solve problems as an individual as well as a member of a team.

Methods of Evaluation:

1. Class participation and discussion
2. Oral and/or written reports
3. Homework assignments
4. Comprehensive projects
5. Quizzes
6. Objective examinations
7. Other methods deemed appropriate by the department

Course Content Outline:

1. An overview of computers, logic and the development process
   a. What is a computer program
   b. What is a compiler
   c. Executing an application
   d. What is a variable
   e. What is a statement
   f. Communicating amongst classes
   g. Object-oriented design
   h. Object-oriented vs. structured design
   i. Designing with software tools
   j. Testing the design
   k. Testing the implementation (code)
   l. The software development team
   m. Taking initiative and working as an individual self-starter
2. Object-oriented programming concepts
   a. What is a class
   b. What is an object
   c. The difference between classes and objects
   d. Communicating amongst classes
   e. Object-oriented design
   f. Object-oriented vs. structured design
   g. The use of classes in modularization
   h. Teamwork and modularization
3. Using methods and parameters
   a. Determining program control
   b. Determining the input/output process
   c. Invoking a method
   d. Passing information in a method
   e. Returning a value from a method
   f. The use of methods in modularization
4. Understanding structure
   a. Sequence
   b. Selection (conditions/*if*-statements)
   c. Iteration (loops)
5. Making decisions (*if*-statements)
   a. Simple *if*-statements
   b. *If-then-else* statements
   c. Nested *if*-statements
   d. Determining the appropriate conditional (*if)* structure
   e. Switch/case statements
6. Looping

   a. While loop
   b. Until Loop
   c. For Loop
   d. Nested *if*-statements
   e. Determining the appropriate loop structure
7. Arrays
   a. Single dimensional arrays
   b. Multi-dimensional arrays
   c. Arrays and loops (searching)
   d. The link between arrays and algorithms (i.e. searches, etc.)
8. Advanced object-oriented concepts
   a. Inheritance
   b. Encapsulation
   c. Composition
   d. Polymorphism
   e. Extensibility (anticipating and adapting to change)

Resources

M. Weisfeld. (2019) *The Object-Oriented Thought Process*, Addison-Wesley.

P. Dietel, H. Dietel. (2016) *Visual C# How to Program*, Pearson.

J. Farrell. (2017) *Microsoft Visual C#: An Introduction to Object-Oriented Programming*, Cengage Learning.

T. Gaddis. (2023) *Starting Out with Programming Logic and Design*, Pearson.

(2023) *C# Tutorial*, https://www.w3schools.com/cs/

Introduction to C#. Microsoft, 12/10/2022. https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials/

Resources Other

MSDN Subscriptions
http://msdn.microsoft.com/en-us/subscriptions/default.aspx

Top of page
Key: 2477