IT-1051: INTRODUCTION TO PROGRAMMING

Cuyahoga Community College

Viewing: IT-1051 : Introduction to Programming

Board of Trustees:
January 2026

Academic Term:
Fall 2026

Subject Code
IT - Information Technology

Course Number:
1051

Title:
Introduction to Programming

Catalog Description:
Students learn to solve business problems by designing, coding, and testing programming solutions in high-level programming languages. Emphasis is placed on language constructs, control flow, and introductory object-oriented concepts, with integration of AI-assisted programming tools to enhance learning and practice.

Credit Hour(s):
3

Lecture Hour(s):
2

Lab Hour(s):
2

Requisites

Prerequisite and Corequisite
IT-1026 Introduction to Computing, 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. Labs/Programming Assignments
2. Quizzes
3. Discussions/Participation
4. Exams
5. Projects
6. AI-Augmented Work

Course Content Outline:

1. Orientation & Basics
   a. Problem-solving
   b. Syntax
   c. Program.cs
   d. First run
2. Variables & Data Types
   a. Arithmetic
   b. Input/output
   c. Strings
   d. Constants
3. Control Structures
   a. if/else
   b. Switch
   c. Conditions
   d. Boolean logic
4. Repetition (Loops)
   a. while
   b. for
   c. do..while
   d. Loop tracing
5. Arrays & Lists
   a. Iteration
   b. Searching
   c. Sorting
6. Classes & Objects
   a. Fields, properties, methods
   b. AI Pair Programming with Copilot
7. Methods
   a. Method overloading
   b. Parameters
   c. Return types
8. Constructors & Exception Handling
   a. Intro to debugging with try/catch
9. Debugging
   a. Strategies
   b. Tracing
   c. Error-handling practice
10. Inheritance
    a. Base/derived classes
    b. Method overriding
11. Polymorphism
    a. Basics
    b. Virtual/override
    c. Interfaces
12. Advanced Object-Oriented Programming (OOP) Design

a. Abstraction
b. Encapsulation

13. Testing & Debugging
    a. Unit testing basics
    b. Bug fixing
14. Capstone & Wrap-up
    a. Capstone Project & Final Exam
    b. Course wrap-up

Religious Accommodation

Before reviewing the course schedule, students should carefully review the following religious accommodation policy and other required instructional policies:

Religious Accommodation:

Students seeking an accommodation for absences permitted under Ohio's Testing Your Faith Act must provide the instructor with written notice of the specific dates for which the student requires an accommodation and must do so not later than fourteen (14) days after the first day of instruction. Please submit requests for accommodations at this link: https://portal2.tri-c.edu/ReligiousAccommodation/ Religious Accommodation Form. Students with questions about their religious accommodations under Ohio's Testing Your Faith Act may contact the College's Office of General Counsel and Legal Services by phone at 216.987.4856 or via email at legal@tri-c.edu.

Other Required Instructional Policies:

https://www.tri-c.edu/student-resources/curriculum/documents/syllabus-part-b.pdf

Weekly Schedule

	Topics
Week 1	Orientation & Basics, Problem-solving, Syntax, Program.cs, First run Lab 1
Week 2	Variables & Data Types, Arithmetic, Input/output, Strings, Constants Lab 2
Week 3	Control Structures — if/else, Switch, Conditions, Boolean logic Lab 3
Week 4	Repetition (Loops) — while, for, do..while, Loop tracing Lab 4
Week 5	Arrays & Lists, Iteration, Searching, Sorting, Teamwork Checkpoint #1 Lab 5
Week 6	Classes & Objects — fields, properties, methods, AI Pair Programming with Copilot Lab 6
Week 7	Classes & Objects (continued), Method overloading, Parameters, Return types, Midterm Exam, AI with ChatGPT, Teamwork Checkpoint #2 Lab 7
Week 8	Constructors & Exception Handling, Intro to debugging with try/catch Lab 8 Midterm Exam
Week 9	Debugging (continued), Strategies, Tracing, Error-handling practice Lab 9
Week 10	Inheritance — base/derived classes, Method overriding Lab 10
Week 11	Inheritance (continued), Polymorphism basics, Teamwork Checkpoint #3 Lab 11
Week 12	Polymorphism (continued), Virtual/override, Interfaces Lab 12
Week 13	Advanced OOP Design — abstraction, Encapsulation Lab 13
Week 14	Advanced OOP (continued) Lab 14

Week 15	Testing & Debugging — unit testing basics, Bug fixing, Integration review Lab 15
Week 16	Capstone Project & Final Exam

The Course Schedule is subject to change due to pedagogical needs, instructor discretion, parts of term, and unexpected events.

Required/Recommended Readings

Readings may be selected from the following:

- Access Modifiers – https://www.w3schools.com/cs/cs_access_modifiers.php (https://www.w3schools.com/cs/cs_access_modifiers.php)
- Booleans – https://www.w3schools.com/cs/cs_booleans.php (https://www.w3schools.com/cs/cs_booleans.php)
- C# Comments – https://www.w3schools.com/cs/cs_comments.php (https://www.w3schools.com/cs/cs_comments.php)
- C# Constructor Methods – https://www.w3schools.com/cs/cs_constructors.php (https://www.w3schools.com/cs/cs_constructors.php)
- C# Corner: Advanced OOP concepts in C# – https://www.c-sharpcorner.com/article/object-oriented-programming-using-C-Sharp/ (https://www.c-sharpcorner.com/article/object-oriented-programming-using-C-Sharp/)
- C# Corner: Encapsulation in C# – https://www.c-sharpcorner.com/UploadFile/ff2f08/encapsulation-in-C-Sharp/ (https://www.c-sharpcorner.com/UploadFile/ff2f08/encapsulation-in-C-Sharp/)
- C# Corner: Introduction to Unit Testing in C# – https://www.c-sharpcorner.com/article/unit-testing-using-nunit/ (https://www.c-sharpcorner.com/article/unit-testing-using-nunit/)
- C# Corner: Polymorphism in C# – https://www.c-sharpcorner.com/UploadFile/9b86d4/polymorphism-in-C-Sharp/ (https://www.c-sharpcorner.com/UploadFile/9b86d4/polymorphism-in-C-Sharp/)
- C# Corner: Understanding Inheritance in C# – https://www.c-sharpcorner.com/UploadFile/mahesh/inheritance-in-C-Sharp/ (https://www.c-sharpcorner.com/UploadFile/mahesh/inheritance-in-C-Sharp/)
- C# Corner: Understanding Interfaces in C# – https://www.c-sharpcorner.com/UploadFile/b1df45/interfaces-in-C-Sharp/ (https://www.c-sharpcorner.com/UploadFile/b1df45/interfaces-in-C-Sharp/)
- C# Corner: Working with Arrays – https://www.c-sharpcorner.com/uploadfile/mahesh/working-with-arrays-in-C-Sharp/ (https://www.c-sharpcorner.com/uploadfile/mahesh/working-with-arrays-in-C-Sharp/)
- C# Exception Handling – https://www.w3schools.com/cs/cs_exceptions.php (https://www.w3schools.com/cs/cs_exceptions.php)
- C# Home – https://www.w3schools.com/cs/index.php (https://www.w3schools.com/cs/)
- C# if...else – https://www.w3schools.com/cs/cs_conditions.php (https://www.w3schools.com/cs/cs_conditions.php)
- C# Intro – https://www.w3schools.com/cs/cs_intro.php (https://www.w3schools.com/cs/cs_intro.php)
- C# Methods – https://learn.microsoft.com/en-us/dotnet/csharp/methods (https://learn.microsoft.com/en-us/dotnet/csharp/methods/)
- C# Output – https://www.w3schools.com/cs/cs_output.php (https://www.w3schools.com/cs/cs_output.php)
- C# switch Statement – https://www.w3schools.com/cs/cs_switch.php (https://www.w3schools.com/cs/cs_switch.php)
- C# Tutorial: Intro to arrays – https://www.w3schools.com/cs/cs_arrays.php (https://www.w3schools.com/cs/cs_arrays.php)
- Class Members – https://www.w3schools.com/cs/cs_classes.php#members (https://www.w3schools.com/cs/cs_classes.php#members)
- Class Properties – https://www.w3schools.com/cs/cs_properties.php (https://www.w3schools.com/cs/cs_properties.php)
- Classes & Objects – https://www.w3schools.com/cs/cs_classes.php (https://www.w3schools.com/cs/cs_classes.php)
- Classes and Objects – https://www.w3schools.com/cs/cs_classes.php (https://www.w3schools.com/cs/cs_classes.php)
- Constructor Method Overloading Example – https://www.w3schools.com/cs/cs_constructors.php#overloading (https://www.w3schools.com/cs/cs_constructors.php#overloading)
- Continue and Break – https://www.w3schools.com/cs/cs_break.php (https://www.w3schools.com/cs/cs_break.php)
- Create a C# Class – https://www.w3schools.com/cs/cs_classes.php (https://www.w3schools.com/cs/cs_classes.php)
- Default Parameters – https://www.w3schools.com/cs/cs_method_parameters_default.php (https://www.w3schools.com/cs/cs_method_parameters_default.php)
- For Loop – https://www.w3schools.com/cs/cs_for_loop.php (https://www.w3schools.com/cs/cs_for_loop.php)
- Intro to OOP (W3Schools) – https://www.w3schools.com/cs/cs_oop.php (https://www.w3schools.com/cs/cs_oop.php)
- List Tutorial (optional) – https://www.w3schools.com/cs/cs_lists.php (https://www.w3schools.com/cs/cs_lists.php)
- Method Overloading – https://www.w3schools.com/cs/cs_method_overloading.php (https://www.w3schools.com/cs/cs_method_overloading.php)
- Method Parameters – https://www.w3schools.com/cs/cs_method_parameters.php (https://www.w3schools.com/cs/cs_method_parameters.php)
- Methods (W3Schools) – https://www.w3schools.com/cs/cs_methods.php (https://www.w3schools.com/cs/cs_methods.php)
- Microsoft Learn: Abstract and sealed classes – https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/abstract (https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/abstract/)

- Microsoft Learn: base keyword for accessing parent class members – https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/base (https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/base/)
- Microsoft Learn: Debugging in Visual Studio Code – https://learn.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-debugging?view=vs-2022 (https://learn.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-debugging/?view=vs-2022)
- Microsoft Learn: Debugging Techniques and Tools – https://learn.microsoft.com/en-us/visualstudio/debugger/debugger-feature-tour (https://learn.microsoft.com/en-us/visualstudio/debugger/debugger-feature-tour/)
- Microsoft Learn: Encapsulation (get/set, private fields) – https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/ (https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/)
- Microsoft Learn: Explore data types and variables using C# – https://learn.microsoft.com/en-us/training/modules/csharp-literals-variables/ (https://learn.microsoft.com/en-us/training/modules/csharp-literals-variables/)
- Microsoft Learn: Inheritance in C# – https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/inheritance (https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/inheritance/)
- Microsoft Learn: Interfaces in C# – https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/types/interfaces (https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/types/interfaces/)
- Microsoft Learn: Object-Oriented Programming Overview – https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/ (https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/)
- Microsoft Learn: Organizing classes with namespaces – https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/namespaces/ (https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/namespaces/)
- Microsoft Learn: Polymorphism in C# – https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/polymorphism (https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/polymorphism/)
- Microsoft Learn: Static Classes & Members – https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/static-classes-and-static-class-members (https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/static-classes-and-static-class-members/)
- Microsoft Learn: virtual, override, and base keywords – https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/virtual (https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/virtual/)
- Microsoft Learn: Write unit tests with MSTest – https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-mstest (https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-mstest/)
- Microsoft Tutorial: Writing your first C# code – https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials/ (https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials/)
- Multiple Objects – https://www.w3schools.com/cs/cs_classes.php#multiple (https://www.w3schools.com/cs/cs_classes.php#multiple)
- Named Arguments – https://www.w3schools.com/cs/cs_method_parameters_named.php (https://www.w3schools.com/cs/cs_method_parameters_named.php)
- Operators (Arithmetic, Assignment, Comparison, Logical) – https://www.w3schools.com/cs/cs_operators.php (https://www.w3schools.com/cs/cs_operators.php)
- Parameters – https://www.w3schools.com/cs/cs_methods.php#parameters (https://www.w3schools.com/cs/cs_methods.php#parameters)
- Return Values – https://www.w3schools.com/cs/cs_method_return.php (https://www.w3schools.com/cs/cs_method_return.php)
- Selection Statements – https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/statements/selection-statements (https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/statements/selection-statements/)
- W3Schools: C# Abstract Classes – https://www.w3schools.com/cs/cs_abstract.php (https://www.w3schools.com/cs/cs_abstract.php)
- W3Schools: C# Access Modifiers – https://www.w3schools.com/cs/cs_access_modifiers.php (https://www.w3schools.com/cs/cs_access_modifiers.php)
- W3Schools: C# Constants – https://www.w3schools.com/cs/cs_constants.php (https://www.w3schools.com/cs/cs_constants.php)
- W3Schools: C# Data Types – https://www.w3schools.com/cs/cs_data_types.php (https://www.w3schools.com/cs/cs_data_types.php)
- W3Schools: C# Exceptions (review before debugging) – https://www.w3schools.com/cs/cs_exceptions.php (https://www.w3schools.com/cs/cs_exceptions.php)
- W3Schools: C# Identifiers – https://www.w3schools.com/cs/cs_identifiers.php (https://www.w3schools.com/cs/cs_identifiers.php)
- W3Schools: C# Inheritance – https://www.w3schools.com/cs/cs_inheritance.php (https://www.w3schools.com/cs/cs_inheritance.php)
- W3Schools: C# Interface – https://www.w3schools.com/cs/cs_interface.php (https://www.w3schools.com/cs/cs_interface.php)
- W3Schools: C# Math – https://www.w3schools.com/cs/cs_math.php (https://www.w3schools.com/cs/cs_math.php)
- W3Schools: C# Method Overriding – https://www.w3schools.com/cs/cs_polymorphism.php#method-override (https://www.w3schools.com/cs/cs_polymorphism.php#method-override)
- W3Schools: C# Multiple Variables – https://www.w3schools.com/cs/cs_variables_multiple.php (https://www.w3schools.com/cs/cs_variables_multiple.php)
- W3Schools: C# Namespaces – https://www.w3schools.com/cs/cs_namespaces.php (https://www.w3schools.com/cs/cs_namespaces.php)

- W3Schools: C# OOP — Abstraction – https://www.w3schools.com/cs/cs_abstract.php (https://www.w3schools.com/cs/cs_abstract.php)
- W3Schools: C# Operators – https://www.w3schools.com/cs/cs_operators.php (https://www.w3schools.com/cs/cs_operators.php)
- W3Schools: C# Polymorphism – https://www.w3schools.com/cs/cs_polymorphism.php (https://www.w3schools.com/cs/cs_polymorphism.php)
- W3Schools: C# Polymorphism (intro to override) – https://www.w3schools.com/cs/cs_polymorphism.php (https://www.w3schools.com/cs/cs_polymorphism.php)
- W3Schools: C# Properties (encapsulation) – https://www.w3schools.com/cs/cs_properties.php (https://www.w3schools.com/cs/cs_properties.php)
- W3Schools: C# Static Classes & Members – https://www.w3schools.com/cs/cs_classes_static.php (https://www.w3schools.com/cs/cs_classes_static.php)
- W3Schools: C# String Concatenation – https://www.w3schools.com/cs/cs_strings_concat.php (https://www.w3schools.com/cs/cs_strings_concat.php)
- W3Schools: C# String Interpolation – https://www.w3schools.com/cs/cs_string_interpolation.php (https://www.w3schools.com/cs/cs_string_interpolation.php)
- W3Schools: C# Strings – https://www.w3schools.com/cs/cs_strings.php (https://www.w3schools.com/cs/cs_strings.php)
- W3Schools: C# Type Casting – https://www.w3schools.com/cs/cs_type_casting.php (https://www.w3schools.com/cs/cs_type_casting.php)
- W3Schools: C# User Input – https://www.w3schools.com/cs/cs_user_input.php (https://www.w3schools.com/cs/cs_user_input.php)
- W3Schools: C# Variables – https://www.w3schools.com/cs/cs_variables.php (https://www.w3schools.com/cs/cs_variables.php)
- W3Schools: Variables – https://www.w3schools.com/cs/cs_variables.php (https://www.w3schools.com/cs/cs_variables.php)
- While Loop – https://www.w3schools.com/cs/cs_loops.php (https://www.w3schools.com/cs/cs_loops.php)

Resources for the Instructor

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/

Additional Resources for the Instructor

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

Instructional Services

OAN Number:

Transfer Assurance Guides OCS001 and OIT001

CTAN Number:

Career Technical Assurance Guides CTPROG001 and CTPROG005

Top of page

Key: 5400