

2667-Introduction to Programming

Introduction

Elements of this syllabus are subject to change.

This three-day, instructor-led course introduces students to computer programming. Students will learn the fundamental concepts and terminology of software application development and develop skills in designing and writing simple computer programs. The course assumes no programming background and provides an overview of the software development process in addition to introducing important programming constructs and methodologies.

The course covers such topics as programming language characteristics, integrated development environments, flowcharts, algorithms and pseudocode, variables, operators, conditional statements, looping statements, procedures, error-handling and debugging, object-oriented programming techniques, user interface design, software modeling, and Extensible Markup Language (XML) Web services.

[↑Top of page](#)

Audience

This course is intended for students who wish to:

- Learn what software development is and what software developers do.
- Learn programming concepts and terminology to facilitate communication with software developers.
- Learn to read, trace, and understand simple code.
- Learn to write, test, and debug code to solve a simple problem.
- Evaluate their personal aptitude for career as a programmer or software developer.

This course is appropriate for:

- Individuals who are contemplating a career in software development and who wish to understand the fundamentals of computer programming.
- Professionals, such as managers and technical salespeople, who must understand programming concepts and be able to communicate with software developers.
- End-users of productivity applications, such as Microsoft Word, Microsoft Excel, Microsoft PowerPoint®, and Microsoft Access, who want to add advanced functionality to documents and databases.
- Web designers who want to understand how programs and scripts can make Web sites more dynamic.

[↑Top of page](#)

At Course Completion

After completing this course, students will be able to:

- Describe a typical computer system and its critical components.
- Describe the software development process, its purpose, critical steps, and where programming fits in that process.
- Describe the purposes of programming and software development.
- Describe the evolution of common characteristics of, and differences among, modern

programming languages.

- Define an integrated development environment.
- Describe the architectural aspects of a software application.
- Identify a problem that requires a programmed solution.
- Use common abstract methods to describe the solution concept.
- Implement variables.
- List and describe common operators.
- Implement conditional statements.
- Implement looping constructs.
- Implement procedures.
- Handle errors.
- Implement object-oriented programming techniques.
- Model an object-oriented program.
- Build a Windows-based application.
- Build a Web application.
- Describe XML Web services.

[↑Top of page](#)

Prerequisites

This course assumes students have these skills:

- Basic familiarity with using a computer, such as browsing the Internet or using a word-processing application.
- The ability to use a mouse and keyboard to navigate through the Microsoft Windows® user interface.
- The ability to perform simple file access tasks, such as browsing a directory structure, opening and saving files, and creating folders.

[↑Top of page](#)

Course Materials

The student kit includes a comprehensive workbook and other necessary materials for this class.

[↑Top of page](#)

Course Outline

Module 1: Introduction to Computer Programs

This module introduces the software development process and fundamentals of computer software. Students will learn how computer programs execute, and the types of programs that developers are generally asked to build. They will also learn about different programming languages, techniques, and technologies used by developers in the software development process.

Lessons

- Overview of Software Development
- Phases in the Execution of a Computer Program
- Overview of Problem-Solving Techniques

- Overview of Program Types

Lab 1: Determining the Program Phases and Examining Them in Code

- Determining Program Phases
- Finding the Program Phases in Code

After completing this module, students will be able to:

- Identify the three phases in the execution of a computer program.
- Define computer programs and programming languages.
- Describe problem-solving techniques.

Module 2: Introduction to Scripting

This module introduces the concept of scripting, and shows how scripting can be used on the Web, to create macros, and in Windows. Students will also learn about Hypertext Markup Language (HTML) and its use on the Web.

Lessons

- Using Web Scripting
- Using Macro Scripting
- Using Windows Scripting

Lab 2: Writing Scripts

- Creating "Hello World" with a Web Script
- Creating "Hello World" with a Macro
- Creating "Hello World" with a Windows Script

After completing this module, students will be able to:

- Define basic Web terms, including HTML, Web browser, Internet, client, and server.
- Create a simple Web page by using Notepad.
- Identify the use of Web scripts, macros, and Windows scripts.
- Describe how Web scripts work.
- Describe how macros work.
- Describe how Windows scripts work.

Module 3: Using Data and Variables

This module introduces the fundamental concepts of variables and data types, which are common to almost all programming languages. Students will learn how to implement variables to store data and how to use operators to manipulate data and variables.

Lessons

- Using Variables
- Using Operators
- Using Programming Syntax

Lab 3: Using Arithmetic Operators

- Calculating the Annual Remuneration of an Employee

After completing this module, students will be able to:

- Explain how to use variables.

- Explain how to use operators.
- Describe the various data types you can use to declare variables.

Module 4: Using Program Logic

This module introduces common programming constructs used in software development to control program flow. Students will learn how to implement conditional expressions and looping statements. They will also learn how to identify errors that can occur in computer programs as a result of incorrect program flow.

Lessons

- Using Branching
- Using Loops
- Identifying Logic Errors

Lab 4: Selecting Candidates Based on Interview Results

- Writing the Pseudocode

After completing this module, students will be able to:

- Explain how to use the IF construct.
- Explain how to use FOR and WHILE looping constructs.
- Define and identify logic errors.

Module 5: Using Procedures and Functions

This module introduces the use of procedures and functions to create more modular computer programs. Students will learn about arguments and parameters and how functions return values.

Lessons

- Using Procedures
- Using Functions

Lab 5: Identifying Functions and Passing Arguments to a Procedure

- Identifying Functions
- Passing Arguments to a Procedure

After completing this module, students will be able to:

- Explain how to use procedures.
- Explain how to use functions.

Module 6: Introduction to Developing a User Interface

This module explains how to design user interfaces for software applications. Students will learn best practices for user interface design and learn how user interfaces are created in Microsoft Visual Studio® .NET.

Lessons

- Designing a User Interface
- Building a User Interface

Lab 6: Building a User Interface

- Building a Web Interface

After completing this module, students will be able to:

- Identify the features of a good user interface.
- Build Windows-based and Web-based user interfaces in Visual Studio .NET.

Module 7: Introduction to Working with Data

This module introduces the concepts of data storage and management. Students will learn about relational data, entity relationships, and primary and foreign keys. They will also learn how XML can be used to describe and structure data.

Lessons

- Introduction to Managing Data
- Using the Entity Relationship Model
- Sorting Data by Using Keys
- Using XML

After completing this module, students will be able to:

- Describe the methods of storing and managing data.
- Describe entity relationship diagrams.
- Identify the various keys used for sorting data.
- Describe XML.

Module 8: Programming Approaches

This module introduces common programming methodologies and compares procedural programming to object-oriented programming. Students will learn the advantages afforded by object-oriented programming techniques. They will also learn how classes of objects can be defined.

Lessons

- Using the Procedural Programming Approach
- Using the Object-Oriented Programming Approach

Lab 8: Examining Classes

- Identifying Classes

After completing this module, students will be able to:

- Describe the two approaches to programming.
- Identify the features and advantages of object-oriented programming.

Module 9: Introduction to the Software Development Process

This module introduces the four phases of the software development lifecycle, and explains the use of the Unified Modeling Language (UML) to identify use cases and to simplify software design.

Lessons

- Software Development Phases
- Introduction to the Unified Modeling Language
- Developing Use-Case Diagrams
- Examining Other UML Diagrams

Lab 9: Developing a Use-Case Diagram

- Identifying Use Cases

- Creating a Use-Case Diagram

After completing this module, students will be able to:

- Identify the four phases in the software development life cycle.
- Identify the views in UML.

Module 10: Where Do We Go From Here?

This module reviews material covered in the course and introduces advanced topics that students may wish to pursue. Students will learn how to apply the skills and knowledge obtained in the course to construct a software solution. They will also learn what options are available for pursuing a career in software development.

Lessons

- Looking Back
- Looking Forward

Lab 10: Putting Programming Fundamentals Together

- Developing a Windows-based Application
- Developing a Web-based Application

After completing this module, students will be able to:

- Identify types of programs.
- Identify areas for advanced training.