

Principles of Programming

Audience	This class is for those students who will be taking a programming class in a specific programming language. It introduces students to some of the general principles of programming. These principles are language independent. This is an appropriate first class for a student planning to learn a computer programming language.
Description	<p>This course uses a combination of lecture, workshops, and exercises to familiarize students with the steps necessary to accomplish the analysis and design of a computer program. Students are introduced to the many phases of the computer programming model. Generic tools are explained for use in solving complex problems. Workshops with exercises are provided to allow students to employ the skills just learned.</p> <p>Students follow two case studies through the class, one for each day. These case studies are designed to introduce students to all of the concepts covered in class. Workshops with exercises are oriented around the ongoing development of the case studies.</p>
Objectives	<p>After completing the class, the students should be able to:</p> <ul style="list-style-type: none">■ understand and work with generic programming constructs,■ discuss problem definition concepts,■ analyze and deconstruct problems into their component parts,■ design programming solutions to problems,■ use flow charts and pseudocode to model programming solutions,■ list data types (numbers, text, date, boolean),■ illustrate the use of array concepts and principles,■ understand file concepts, including input-from and output-to files,■ define the meaning of a computer language,■ document the programming solution,■ describe the use of system and subsystem testing of programs,■ illustrate user interface concepts.
Length	2 days
Beyond this Level	When you have successfully completed this class, the next class you should consider is a Visual Basic for Applications programming class. Using VBA will give you greater flexibility in applications such as Word, Excel, and Access. These classes will help you to maximize your productivity and efficiency when using these applications.



Day 1 (a.m.):

I. What is a Program?

- A. Introduction
 - 1. Hardware
 - 2. Software
 - 3. Programming
 - 4. Programs
 - 5. Stages of Development
- B. Sequence Constructs
- C. Decision Constructs
 - 1. If-Then
 - 2. Case Statement
- D. Loop Constructs
 - 1. For Loop
 - 2. Do While Loop
 - 3. Do Until Loop
- E. Subroutine Constructs

II. Problem Analysis and Solution Analysis

- A. Problem Definition
 - 1. Problem Analysis
 - 2. Problem Deconstruction
 - a. Tasks
 - b. Subtasks
- B. Feasibility Study
- C. Solution Analysis

III. Problem Solution

- A. Solution Designing
- B. Design Tools
- C. Revisiting Problem Analysis
- D. Solution Implementation
- E. Maintenance

Day 1 (p.m.):

IV. Flowcharts

- A. Symbol Set Definition
- B. Sequences
- C. Decision Blocks
- D. Loops
- E. Subroutines
- F. Rules of Flowcharting

V. PseudoCode

- A. Sequences
- B. Decision Structures
- C. Loop Structures
- D. Subroutines

Day 2 (a.m.):

VI. Variables and Data Types

- A. Introduction
- B. Number Variables
- C. Text Variables
- D. Boolean Variables
- E. Date Variables
- F. Operators

VII. File Access

- A. Introduction
- B. File Input
- C. File Output

VIII. Arrays

- A. Introduction
- B. Dimensions and Elements
- C. Input/Output

Day 2 (p.m.):

IX. Coding & Documentation

- A. Computer Languages
- B. Coding
- C. Documenting the Program

X. User Interface

- A. Introduction
- B. Input/Output Considerations
- C. Design Concepts

XI. Testing, Debugging & Maintenance

- A. Introduction
- B. Subsystem Testing
- C. Full System Testing
- D. Error Types
- E. Debugging
- F. Maintenance
- G. Course Summary/Synthesis