

Computer Programming 1

Computer Programming I

Levels:	10-12
Units of Credit:	1.0
CIP Code:	11.0201
Core Code:	35-02-00-00-030
Prerequisites:	Secondary Math I, Keyboarding Proficiency, Computer Literacy requirement
Semester 1 Skill Test:	#820 Computer Programming 1A
Semester 2 Skills Tests:	#822 Computer Programming IB (C++) #824 Computer Programming IB (Java) #826 Computer Programming IB (VB) #827 Computer Programming IB (Python) #828 Computer Programming IB (C#)

COURSE DESCRIPTION

An introductory course in computer programming/software engineering and applications. The course introduces students to the fundamentals of computer programming. Students will learn to design, code, and test their own programs while applying mathematical concepts. Teachers introduce concepts and problem solving skills to beginning students through a programming language such as C++, C#, Java, Python, or VB.

The second half of the year reviews and builds on the concepts introduced in the first semester. This semester introduces students to more complex data structures and their uses, including sequential files, arrays, and classes. Students will learn to create more powerful programs.

(*Semester 2 objectives)

CORE STANDARDS, OBJECTIVES AND INDICATORS

STANDARD 1

Students will be familiar with and use a programming environment.

Objective 1: Demonstrate knowledge of external and internal computer hardware.

- Describe the functions of basic external computer hardware devices (monitor, printer, keyboard, mouse, adapters, other devices).
- Describe the functions of the internal components of computers (CPU, RAM, ROM, motherboard, graphics card, hard drive, optical drive).
- Understand what a bit and a byte is and how it relates to memory storage.

Objective 2: Demonstrate knowledge of software concepts.

- Define the distinction between computer software and hardware.
- Identify software categories such as application software, web-based software, or OS.
- Describe the difference between an interpreted language vs a compiled language.

Objective 3: Demonstrate the ability to compile, debug, and execute programs.

- Demonstrate how to use an editor/IDE to compile and run programs.
- Understand the difference between syntax, run-time, and logic errors.
- Demonstrate how to debug programs.
- Optional -- Use a debugger to set break-points, and step through code to track down errors at runtime.

STANDARD 2

Students will employ accepted programming methodology.

Objective 1: Demonstrate the ability to use good programming style.

- Demonstrate how to use white space properly.
- Employ an appropriate naming convention.

Computer Programming 1

- c. Construct identifiers with meaningful format (ie: camelCase, under_scores,PascalCase, and ALLCAPS).

Objective 2: Understand that software development is a process and use a variety of creation techniques to develop 21st Century Skills.(www.p21.org)

- a. Understand specifications and requirements for computer programs.
- b. Decompose the problem into appropriate components.
- c. Design solutions using algorithms and other problem solving techniques..
- d. Write the code for a program.
- e. Test programs for errors and proper functionality.
- f. Provide internal and external documentation for a program during development.
- g. Redo all steps as needed.

Objective 3: Identify the syntactical components of a program.

- a. Identify keywords, identifiers, operators, operands, and literals.
- b. Identify the entry-point of a program.
- c. Identify statements and expressions in a program.
- d. Identify program components such as functions, methods, or procedures.

STANDARD 3

Students will properly use language-fundamental commands and operations.

Objective 1: Demonstrate the ability to use basic elements of a specific language.

- a. Write programs formatted based on the conventions of the utilized language.
- b. Declare, initialize, and assign values to constants and variables.
- c. Demonstrate the ability to use input and output commands.
- d. Communicate clearly with output values stored in identifiers.(www.p12.org)
- e. Demonstrate the ability to use strings in programs.

Objective 2: Employ basic arithmetic expressions in programs.

- a. Use basic arithmetic operators (modulus, multiplication, division, addition, subtraction).
- b. Understand order of operation of expressions.
- c. Write expressions that mix floating-point and integer expressions.

Objective 3: Demonstrate the ability to use data types in programs.

- a. Declare and use variable types (primitives, reference, or object).
- b. Declare and use constants.
- c. Know the difference between data types and their application (boolean, integer, floating point, strings).
- d. Optional -- Declare and use enumerators as a list of constants.

STANDARD 4

Students will properly employ control structures.

Objective 1: Demonstrate the ability to use relational and logical operators in programs.

- a. Compare values using relational operators.
- b. Form complex expressions using logical operators.

Objective 2: Demonstrate the ability to use decisions in programs.

- a. Employ simple IF structures.
- b. Use IF-ELSE structures.
- c. Write programs with nested IF-ELSE structures.
- d. Make multiple-way selections (switch, case).

Objective 3: Demonstrate the ability to use loops in programs.

- a. Use initial, terminal, and incremental values in loops.
- b. Construct while, do-while, and for loops

Computer Programming 1

- c. Describe the various ways that loops can end.
- d. Utilize nested loops.
- e. Explain how to avoid infinite loops.
- f. Accumulate running totals using loops.

Objective 4: Demonstrate the ability to use modularity in programs using functions or methods.

- a. Demonstrate how to use language-defined components.
- b. Utilize value and reference parameters.
- c. Understand the scope of identifiers (local, class variables).
- d. Return values.

STANDARD 5

Students will demonstrate knowledge of current ethical issues dealing with computers and information in a global society using 21st Century Skills.

Objective 1: Understand ethical responsibility of software developers

- a. Explain the ethical reasons for creating reliable and robust software.
- b. Explain the impact software can have on society.
- c. Show how security concerns can be addressed in a program.

Objective 2: Demonstrate knowledge of the social and ethical consequences of computers.

- a. Describe how computer-controlled automation affects a workplace and society.
- b. Explain the ramifications of society's dependence on computers.
- c. Use 21st Century Skills to understand and address global issues
- d. Identify advantages and disadvantages of changing workplace environments.
- e. Be aware of changing tools in technology and adapt to a changing environment.

Objective 3: Demonstrate knowledge of the right to privacy.

- a. Explain how computers can compromise privacy.
- b. Exhibit knowledge of privacy laws.
- c. Describe responsibilities of people who control computer information.

Objective 4: Demonstrate knowledge of computer, information and software security.

- a. Exhibit knowledge of copyright laws.
- b. Explain how computers could erroneously be used to compromise copyright laws.
- c. Give examples of ways to protect information on computer systems.
- d. Identify ways to protect against computer viruses.

STANDARD 6

Students will develop an awareness of career opportunities in the Computer Programming/Software Engineering industry and of its history.

Objective 1: Identify personal interests and abilities related to Computer Programming/Software Engineering careers

- a. Identify personal creative talents
- b. Identify technical/programming talents
- c. Identify organizational and leadership skills
- d. Explore aptitude for innovation
- e. Determine aptitude for working as a member of a computer programming/software engineering team

Objective 2: Investigate career opportunities, trends, and requirements related to computer programming/software engineering careers

- a. Identify the members of a computer programming/software engineering team: team leader, analyst, senior developer, junior developer, and client/subject matter expert
- b. Describe work performed by each member of the computer programming/software engineering team
- c. Investigate trends associated with computer programming/software engineering careers
- d. Discuss related career pathways.

Computer Programming 1

- e. Compile a portfolio of the individual and group programs developed during the course

Objective 3: Discuss relevant history of software development

- a. Discuss relevant history of computer technology
- b. Identify key points in the history of the computer programming/software engineering industry

(Semester 2 Standards*)

STANDARD 7*

Students will employ arrays.

Objective 1: Demonstrate the ability to use arrays in programs.

- a. Declare arrays of all applicable types.
- b. Initialize arrays.
- c. Perform data input to and output from arrays.
- d. Perform operations on arrays including sequential searches.
- e. Iterate through the structure (i.e. foreach loop)

Objective 2: Demonstrate the ability to use dynamic arrays (i.e. vectors, ArrayLists, or generic lists)

- a. Declare a dynamic array
- b. Add and remove items from the array
- c. Output data from arrays.
- d. Perform operations on arrays.
- e. Iterate through the structure (i.e. foreach loop)

Objective 3: Demonstrate the ability to use strings in programs.

- a. Compare string identifiers.
- b. Find the length of a string.
- c. Copy part or all of string identifiers into other strings.
- d. Concatenate string identifiers.
- e. Locate substring positions.
- f. Insert strings into other strings.

STANDARD 8*

Students will properly employ object-oriented programming techniques.

Objective 1: Demonstrate the ability to use existing classes.

- a. Instantiate objects.
- b. Use object data members.
- c. Use object member functions (methods).

Objective 2: Demonstrate the ability to create user-defined classes.

- a. Create and use data members.
- b. Create a constructor to initialize the data members.
- c. Create and use instance functions (methods).

Objective 3: Demonstrate proper design principles with classes.

- a. Create classes that are well encapsulated (data members private).
- b. Properly use modifiers and accessors (getters and setters).
- c. Understand appropriate private and public modifiers according to program design.

Computer Programming 1

STANDARD 9*

Students will properly use sequential files.

Objective 1: Demonstrate the ability to use sequential files in programs.

- a. Create and initialize sequential files.
- b. Store data to sequential files.
- c. Retrieve data from sequential files.
- d. Update sequential files.

STANDARD 10*

Students will apply appropriate programming skill as an effective member of a team demonstrating the ability to collaborate with others (www.p21.org).

Objective 1: Demonstrate the ability to apply knowledge to a programming project.

- a. Formalize specifications.
- b. Choose proper input parameters.
- c. Choose appropriate data structures and processing.
- d. Design appropriate output.
- e. Use appropriate test data.
- f. Write good documentation.

Objective 2: Demonstrate the ability to use teamwork and collaboration in a programming project.

- a. Divide a project among programmers.
- b. Present work to a group.
- c. Coordinate work with others in the group.
- d. Complete assigned work according to predetermined deadlines.
- e. Participate in a peer performance evaluation.
- f. Demonstrate professionalism in team relationships, communication, timeliness, and attitude.