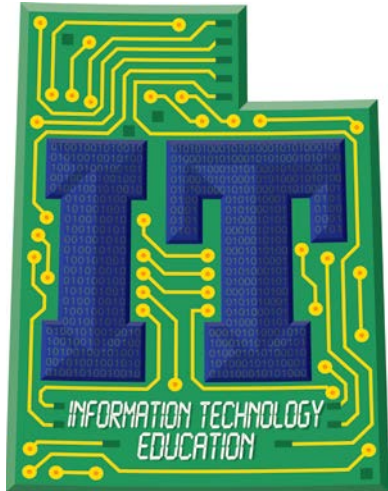


# STRANDS AND STANDARDS

## COMPUTER PROGRAMMING 1



### 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 coding concepts and problem-solving skills to beginning students through a programming language such as C++, C#, Java, Python, or JavaScript.

The course that follows this course is titled Computer Programming 2. This follow-up course reviews and builds on the concepts introduced in the course. This second course introduces students to more complex data structures and their uses, including sequential files, arrays, and classes. Students will learn to create more powerful programs.

(\*Computer Programming 2 explains topics using Language Specific concepts and practices)

<b>Intended Grade Level</b>	9-12
Units of Credit	0.5
Core Code	35.02.00.00.030
Concurrent Enrollment Core Code	35.02.00.13.030
Prerequisite	Suggested - Digital Literacy, Computer Science Principles, or Teacher Approval
Skill Certification Test Number	#820, #9830
Test Weight	0.5
<b>License Type</b>	CTE and/or Secondary Education 6-12
<b>Required Endorsement(s)</b>	
Endorsement 1	Computer Science Level 1 or
Endorsement 2	Computer Science Level 2

## STRAND 1

**Students will be familiar with and use a programming environment.**

### Standard 1

Demonstrate knowledge of software concepts.

- Identify software categories e.g. application software, web-based software, mobile application, or operating system.
- Describe the difference between an interpreted language vs a compiled language.

### Standard 2

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.

### Performance Skills

- Become familiar with and use a programming environment.

## STRAND 2

**Students will employ accepted programming methodology.**

### Standard 1

Demonstrate the ability to use good programming style.

- Demonstrate how to use white space properly.
- Employ an appropriate naming convention.
- Construct identifiers with meaningful format (i.e.: camelCase, underscores, and ALLCAPS).

## Standard 2

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

- Understand specifications and requirements for computer programs.
- Break down the problem into sub-components.
- Design solutions using algorithms and other problem solving techniques.
- Write the code for a program.
- Test programs for errors and proper functionality.
- Provide internal and external documentation for a program during development.
- Redo all steps as needed.

## Standard 3

Identify the syntactical components of a programming language.

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

## Performance Skills

- Employ accepted programming methodology.

## STRAND 3

**Students will properly use language-fundamental commands and operations.**

### Standard 1

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

- Write programs formatted based on the conventions of the utilized language.
- Declare, initialize, and assign values to constants and variables.
- Demonstrate the ability to use input and output commands.
- Communicate clearly with output values stored in identifiers. ([www.p21.org](http://www.p21.org))
- Demonstrate the ability to use strings.

### Standard 2

Employ basic arithmetic expressions in programs.

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

## Standard 3

Demonstrate the ability to use data types in programs.

- Declare and use variable types (primitives, reference, or object).
- Declare and use constants.
- Know the difference between data types and their application (boolean, integer, floating point, strings).

## Performance Skills

- Properly use language-fundamental commands and operations.

## STRAND 4

**Students will properly employ control structures.**

### Standard 1

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

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

### Standard 2

Demonstrate the ability to use decisions in programs.

- Employ simple IF structures.
- Use IF-ELSE structures.
- Write programs with nested IF-ELSE structures.
- Make multiple-way selections (switch, case).\* (Language specific)

### Standard 3

Demonstrate the ability to use loops (iteration) in programs.

- Use initial (starting) value, terminal (ending) condition, and incrementation (change) in loops.
- Construct pretest loops (while), posttest loops (do-while), and for loops.
- Describe the various ways that loops can end (i.e., sentinel, break, condition fail, etc.).
- Design loops so they iterate the correct number of times (i.e., off by one errors, infinite loops, etc.).
- Accumulate running totals using loops.
- Utilize nested loops.

### Standard 4

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

- Demonstrate how to use language-defined functions and/or methods. \*
- Utilize value and/or reference parameters. \*
- Understand the scope of identifiers (local, global (class), and instance variables). \*
- Return values.

## Performance Skills

- Properly employ control structures.

## STRAND 5

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

### Standard 1

Demonstrate knowledge of the social and ethical consequences of computers.

- Explain the ethical reasons for creating reliable and robust software.
- Explain the impact software can have on society (i.e., privacy, piracy, copyright laws, ease of use, etc.).
- Show how security concerns can be addressed in an application (i.e., biometrics, passwords, information hiding, etc.).
- Describe how computer-controlled automation affects a workplace and society.
- Give examples of ways to protect information on computer systems (attacks, viruses, malware, etc.).

## Performance Skills

- Demonstrate knowledge of current ethical issues dealing with computers and information in society.

## STRAND 6 (Optional)

**Students will be aware of career opportunities in the Computer Programming/Software Engineering industry and of its history.**

### Standard 1

Investigate career opportunities, trends, and requirements related to computer programming/software engineering careers.

- Identify the members of a computer programming/software engineering team: team leader, analyst, senior developer, junior developer, and client/subject matter expert.
- Describe work performed by each member of the computer programming/software engineering team.
- Investigate trends and traits associated with computer programming/software engineering careers (creativity, technical, leadership, collaborative, problem solving, design, etc.).
- Discuss related career pathways.

## Performance Skills

- Develop awareness of career opportunities in the computer programming/software engineering industry and of its history.

## STRAND 7 (Optional)

**Students will employ static (array), dynamic (vector, ArrayList, etc.) list structures, and strings. (Semester 2 Strands)**

### Standard 1

Demonstrate the ability to use static arrays/lists in programs.

- Declare and initialize arrays/lists of all applicable types.
- Perform data input to and output from arrays/lists.
- Perform operations on arrays/lists including sort arrays.
- Iterate through the structure (i.e. for-each, enhanced for, or iterators)

### Standard 2

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

- Declare and initialize a dynamic array/list.
- Add and remove items from the array/list.
- Output data from arrays/lists.
- Perform operations on arrays/lists.
- Iterate through the structure (i.e. for-each, enhanced for, or iterators)

### Standard 3

Demonstrate the ability to use strings in programs.

- Compare string values.
- Find the length of a string.
- Copy part or all of string values into other strings.
- Concatenate string values.
- Locate substring positions.
- Insert strings into other strings.

### Performance Skills

- Properly employ static data structures.

## STRAND 8 (Optional)

**Students will properly employ object-oriented programming techniques.**

### Standard 1

Demonstrate the ability to use existing classes.

- Instantiate objects.
- Use object data members (i.e., Java's array.length).
- Use object member functions (methods).

## Standard 2

Demonstrate the ability to create user-defined classes.

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

## Standard 3

Demonstrate proper design principles with classes.

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

## Performance Skills

- Properly employ object-oriented programming techniques.

## STRAND 9 (Optional)

**Students will properly use sequential files.**

### Standard 1

Demonstrate the ability to use sequential files in programs.

- Create and initialize sequential files.
- Store data to sequential files.
- Retrieve data from sequential files.
- Update sequential files.

## Performance Skills

- Properly use sequential files.

## STRAND 10 (Optional)

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

### Standard 1

Demonstrate the ability to apply knowledge to a programming project.

- Formalize specifications.
- Choose proper input parameters.
- Choose appropriate data structures and processing.
- Design appropriate output.
- Use appropriate test data.
- Write good documentation.

## Standard 2

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

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

## Performance Skills

- Apply appropriate programming skills as an effective member of a team.

## Work Place Skills

Communication, Problem Solving, Teamwork, Critical Thinking, Dependability, Accountability, Legal requirements/expectations

## Skill Certificate Test Points by Strand

Test Name	Test #	Number of Test Points by Strand										Total Points	Total Questions
		1	2	3	4	5	6	7	8	9	10		
Computer Programming 1A	820	3	8	14	21	5						51	42
Computer Programming IB (C++)	822	6	2	14	11		1	10	7	6	2	59	50
Computer Programming IB (Java)	824	6	2	14	11		1	10	7	6	2	59	50
Computer Programing IB (VB)	826	6	2	14	11		1	10	7	6	2	59	50
Computer Programming IB (Python)	827	7	2	13	11		1	9	5	1	2	51	51
Computer Programming IB (C#)	828	6	2	14	11		1	10	7	6	2	59	50