Unit 2: Control Structures-Conditionals and Loops

Content Area: **Technology**
Course(s): **Computer Programming C++ I & II**
Time Period: **8 weeks**
Length: **Weeks**
Status: **Published**

Unit Overview

Students will be able to read with understanding and write programs involving conditionals (if-statements) and loops (while loops, for loops, and do-while loops). Students will be able to take a description of an iterative task or algorithm and write code that performs that algorithm. Topics covered include the else keyword and if-else chains, the switch statement, nesting of if statements and looping constructs, the use of break and continue in loops, and the use of flow charts to construct a diagram of the control flow for an algorithm or program.

Transfer

Students will be able to independently use the knowledge acquired in this unit to read and write more involved C++ programs that make use conditional and looping constructs. Additionally, students will know what flow charts are, how they are interpreted, and how to convert a description of a process into a flow chart. Students be able to convert simple flow charts to C++ code, and will understand how algorithms with branching flow charts can be implemented with while loops, for loops and if-statements. Students should be able to use this knowledge to write a programs that perform useful tasks, including the creation of simple games.

Meaning

Understandings

Students will understand...

-that 'if' and 'switch' statements can be used to construct programs with branching control flow

-that 'while', 'do-while' and 'for' loops can be used to construct programs with looping control flow

-how variable scope is relevant to the construction of programs using variables inside code blocks

-the meaning and use of flow charts

Essential Questions

Students will keep considering...

-what is the mental process behind constructing a program that performs a given task?

-how can I identify what variables I will need to express a given algorithm?

-how can I express a given algorithm as C++ code using conditional expressions?

-how can I express a given algorithm as C++ code using looping constructs?

-how can a flow chart be used to visualize and analyze an algorithm or program?

Application of Knowledge and Skill

Students will know...

Students will know...

-that 'if' statements are used to execute a statement only when a given condition (a boolean expression) is true

-that code enclosed in matching curly braces creates a 'code block' or compound statement which allows more than one statement to be executed with a conditional or loop

-that the 'else' keyword is used to execute a separate statement when an if-statement's condition is false

-that variables declared inside a block have a *scope* limited to that block

-the 'switch' statement is used to select which code to execute based on the value of a variable

-'while' loops can be used to execute a block of code repeatedly as long as a given condition is true

-that 'for' loops are equivalent to 'while' loops with a counter variable, and are useful to execute a block of code a specific number of times

-the meaning and use of do-while loops and the use of break and continue in loops

Students will be skilled at...

Students will be skilled at...

-writing programs using if statements, chains of if-else, and the switch statement

-writing programs using while loops, for loops, and do-while loops using the correct syntax

-using break and continue to modify the control flow in a loop

-writing blocks of code using a clear and consistent style and indentation scheme

-reading and interpreting code written by others

-reading, interpreting and constructing flow charts

-creating appropriate algorithms to perform simple stated tasks

Academic Vocabulary

keyword, statement, specific C++ keywords: if, else, switch, break, while, do-while, for, and continue; loop, code block, compound statement, flow chart, control flow, variable scope

Learning Goal 1

Students will be able to read with understanding and write programs using conditionals/ if-statements, identify and correct logical and syntax errors in their own and other's programs.

Target 1

Students will be able to read and write programs that use if-statements to conditionally execute a single statement based on the value of a simple boolean expression involving constants and variables, and predict the results of programs involving if statements.

Additional Standards:

9.3.IT-PRG.1,2,3

TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).

Target 2

Students will be able to read and write programs that use if-statements to conditionally execute a compound statement (a code block), and use more complex boolean expressions, and assess whether a given expression correctly implements a specification.

Additional standards:

9.3.IT-PRG.4,5,6

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

Target 3

Students will be able to read and write programs that use the 'else' keyword and chains of if/else-if statements, and investigate the effect of parentheses grouping on program results.

Additional standards:

9.3.ST.6, 9.3.ST-ET.1,3

TECH.8.2.12.E	Computational Thinking: Programming: Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

Target 4

Students will be able to read and write programs that use the 'switch' statement to execute a sequence of statements selected based on the value of an expression, and use models to illustrate the flow of program logic.

Additional standards:

9.3.ST-ET.4,5

9.3.IT.2

Learning Goal 2

Students will be able to read with understanding and write programs using looping constructs (while, for, and do-while loops), and compare and contrast the use cases for each type of loop.

Target 1

Students will be able to write while loops to execute a simple code block repeatedly as long as a simple condition is true and predict the results of programs involving while loops.

Additional standards:

9.3.IT.13, 9.3.IT-PRG.1, 2

TECH.8.2.12.E	Computational Thinking: Programming: Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

Target 2

Students will be able to write while loops that execute a complex code block that involves variables with scope limited to that block, mastering common patterns of loops, and represent the flow of control in a while loop as a diagram.

Additional standards:

9.3.IT.2

9.3.IT.-PRG.3, 4

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

Target 3

Students will be able to read and write programs that use 'for' loops to concisely represent a counter-controlled loop, and compare and contrast the use cases for 'for' loops versus 'while' loops.

Additional standards:

9.3.IT.13

9.3.ST.6

TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

Target 4

Students will be able to read and write programs that use 'do-while' loops and the 'break' and 'continue' statement inside of loops, and predict the results of programs that use break and continue to modify loop control flow.

Additional standards:

9.3.ST-ET.1, 3, 4

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).

Target 5

Students will be able to read and write programs that use nesting of loops and if-statements to express programs with complex control flow, and select the best loop construct among the various alternatives.

Additional Standards:

9.3.ST-ET.5, 9.3.IT.2, 9.3.IT.13

TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

Learning Goal 3

Students will be able to create and understand flow charts as an aid in visualizing, representing and constructing programs, and think critically about issues of concern in the digital age, taking positions on issues such as managing their digital footprint.

Target 1

Students will be able to draw a flowchart to represent the control flow for a simple program involving conditionals and loops and use it to select the best looping construct or selection structure among the various alternatives.

Additional standards:

9.3.IT-PRG.1, 2

TECH.8.2.12.C.3	Analyze a product or system for factors such as safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, and human factors engineering (ergonomics).
TECH.8.2.12.C.4	Explain and identify interdependent systems and their functions.

Target 2

Students will be able to write a simple program involving conditionals and loops based on a given flowcharted algorithm, predict the results of their own and other's programs.

Additional standards:

9.3.IT-PRG.3, 4

TECH.8.2.12.C.5	Create scaled engineering drawings of products both manually and digitally with materials and measurements labeled.
TECH.8.2.12.C.6	Research an existing product, reverse engineer and redesign it to improve form and function.

Target 3

Students will be able to think critically about important issues of the digital age, e.g. by analyzing their digital footprint and examining censorship policies and related concepts, and taking a position on critical issues.

Additional standards:

9.3.ST-SM.4, 9.3.IT.4

TECH.8.1.12.D.3	Compare and contrast policies on filtering and censorship both locally and globally.
TECH.8.1.12.D.4	Research and understand the positive and negative impact of one's digital footprint.
TECH.8.1.12.D.5	Analyze the capabilities and limitations of current and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs.
TECH.8.1.12.D.CS3	Exhibit leadership for digital citizenship.
TECH.8.1.12.E.1	Produce a position statement about a real world problem by developing a systematic plan of investigation with peers and experts synthesizing information from multiple sources.

Summative Assessment

-Chapter and Benchmark assessments

-Project-based assessments. Possible examples include:

-gradebook management program

-contact list management program

-Space Invaders clone

21st Century Life and Careers

CAEP.9.2.12.C.1	Review career goals and determine steps necessary for attainment.
CAEP.9.2.12.C.2	Modify Personalized Student Learning Plans to support declared career goals.
CAEP.9.2.12.C.7	Examine the professional, legal, and ethical responsibilities for both employers and employees in the global workplace.

Formative Assessment and Performance Opportunities

Daily challenge tasks, individual and group project presentations, portfolio assessments, in-class observation, demonstration and questioning

Accommodations/Modifications

-Supplemental project guides

-One-on-one assistance

-Reverse-Polish Notation Calculator (challenge project)

-Turing Machine project (enrichment)

Unit Resources

Relevant Youtube videos, e.g.:

"Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion"

"Open Networks and Open Society: The Relationship between Freedom, Law, and Technology"

Interdisciplinary Connections

LA.RST.11-12.3	Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
LA.RST.11-12.4	Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11-12 texts and topics.
MA.K-12.1	Make sense of problems and persevere in solving them.
MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.4	Model with mathematics.