* **Loop:** A structure that allows repeated execution of a block of statements
* **Loop body:** A block of statements, Executed repeatedly
* **Iteration:** One execution of any loop
* **while:** The loop-controlling Boolean expression is the first statement
* **for:** A concise format in which to execute loops
* **do…while:** The loop-controlling Boolean expression is the last statement
* **while loop:** Executes a body of statements continually
  + As long as the Boolean expression that controls entry into the loop continues to be true
* Consists of: The keyword while
* **Definite loop:** Performs a task a predetermined number of times
  + Also called a counted loop, consider a for structure
* Write a definite loop: Initialize the loop control variable
* While the loop control variable does not pass a limiting value, the program continues to execute the body of the while loop
* **Infinite loop**
  + A loop that never ends
  + Can result from a mistake in the while loop
  + Do not write intentionally
* Break Out: Suspect an infinite loop when:
  + The same output is displayed repeatedly
  + The screen remains idle for an extended period of time
* **Pitfall**: Failing to Alter the Loop Control Variable Within the Loop Body
  + Prevent the while loop from executing infinitely
* **Pitfall**: Unintentionally Creating a Loop with an Empty Body
  + Loop control variable: A variable that is altered and stored with a new value loopCount = loopCount + 1
  + The equal sign assigns a value to the variable on the left
  + The variable should be altered within the body of the loop
* **Empty body:** A body with no statements, Caused by misplaced semicolons
* **Incrementing the variable**
  + Alter the value of the loop control variable by adding 1
* **Decrementing the variable**
  + Subtract 1 from the loop control variable
* **Clearest and best method**
  + Start the loop control variable at 0 or 1
  + Increment by 1 each time through the loop
  + Stop when the loop control variable reaches the limit
* **Indefinite** **loop**
  + Altered by user input
    - Controlled by the user
    - Executed any number of times
  + Validating data
    - Ensure a value falls within a specified range
    - Use indefinite loops to validate input data
    - If a user enters incorrect data, the loop repeats
* **Prefix ++:** The result is calculated and stored, Then the variable is used
* **Postfix ++**: The variable is used (actually a copy is made), Then the result is calculated and stored
* **for loop:** Used when a definite number of loop iterations is required
  + One convenient statement indicates:
  + The starting value for the loop control variable
  + The test condition that controls loop entry
  + The expression that alters the loop control variable
* **do…while loop**: A **posttest loop**
  + Checks the value of the loop control variable
  + At the bottom of the loop
  + After one repetition has occurred
  + Performs a task at least one time
  + You are never required to use this type of loop
  + Use curly braces to block the statement
* **Nested Loop Logic**
  + Improving Loop Performance
    - Make sure a loop does not include unnecessary operations or statements
    - Consider the order of evaluation for short-circuit operators
    - Make comparisons to zero (0)
    - Employ loop fusion to combine loops
* **Loop fusion**
* A technique of combining two loops into one
* Will not work in every situation
* Using Prefix Incrementing Rather than Postfix Incrementing
* Prefix incrementing method
* ++x
* When the method receives a reference to x, the value is increased and the increased value is returned
* Postfix incrementing method
* x++
* When the method receives a reference to x, a copy of the value is made and stored
* The value is incremented as indicated by the reference
* The copy is returned
* The extra time spent copying causes postfix incrementing to take longer

# Summary\_1

## Execute the while loop

### Initialize the loop control variable, test in the while statement, and alter the loop control variable

## Prefix ++ and postfix ++

### Increase a variable's value by 1

### The variable is used

#### The result is calculated and stored

## Unary operators

### Use with one value

# Summary\_2

## Binary operators

### Operate on two values

## Shortcut operators +=, -=, \*=, and /=

### Perform operations and assign the result in one step

## for loop

### Initializes, tests, and increments in one statement

## do…while loop

### Tests a Boolean expression after one repetition

## Improve loop performance

### Do not include unnecessary operations or statements