while Loop in C++

In this lesson, you will get acquainted with the while loop and its basic syntax.

We'll cover the following

Introduction

Suppose you have $20, and the price of the ice-cream is $5. You want to keep buying the ice-cream until you have no money left. This task is repetitive, and you don't know in advance how many ice-creams you can buy.

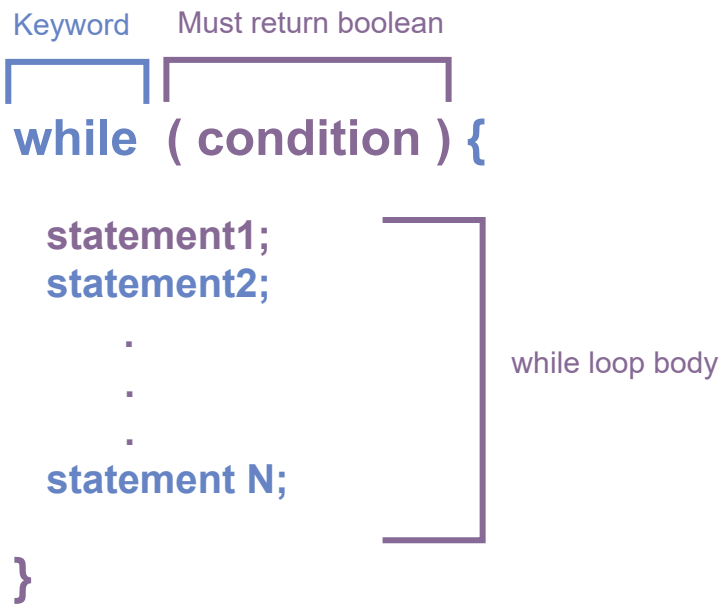
In the era of programming, we can use the `while` loop to implement repetitive tasks.

*The **while loop** keeps executing a particular code block until the given condition is true. It does not know in advance how many times the loop body should be executed.*

> 📝 The condition in the `while` loop is evaluated before executing the statements inside its body. Therefore, the `while` loop is called an entry-controlled loop.

Syntax

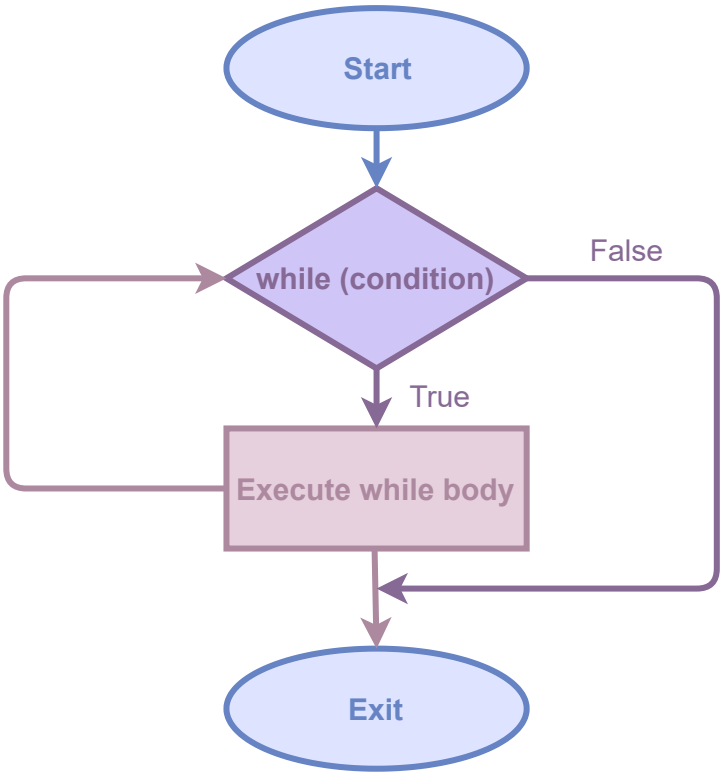
Let's go over the syntax of the `while` loop.



The general syntax of the while loop consists of a `while` keyword followed by a condition to be checked. The closing curly bracket is proceded by the `while` keyword and the condition to be checked.

Flowchart

Let's look at the flowchart of the `while` loop.



- The while loop first evaluates the given condition.
- If the condition evaluates to true, the code inside the body of the `while loop` is executed.
- After that, the `while loop` again evaluates the condition. This process continues until the given condition remains true.

Example program

Let's translate the example given above into a C++ program.

Press the **RUN** button and see the output!

```
#include <iostream>

using namespace std;

int main() {
  // Initialize the variable money
  int money = 20;
  // Initialize the variable icecream_price
  int icecream_price = 5;
  // Prints value of variables
  cout << "Intial money = " << money << endl;
  cout << "Ice-cream price = " << icecream_price << endl;
  // Start of the  while loop
 while (money >= icecream_price){
    // Body of the while loop
    cout << "Buy an ice-cream" << endl;
```

```
    cout << "Buy an ice-cream" << endl;
    money = money - icecream_price;
    cout << "Remaining money = " << money << endl;
  }
  // End of the  while loop
  cout << "You can't buy an ice-cream" << endl;

  return 0;
}
```

Explanation

Line No. 7: Initializes the value of `money`

Line No. 9: Initializes the value of `icecream_price`

Line No. 11: Prints the value of `money` to the console.

Line No. 12: Prints the value of `icecream_price` to the console

Line No. 14: Checks if the value of `money` is greater than or equal to `icecream_price`. If true, then execute **Lines No. 16 to 19**. If false, then it executes **Line No. 21**.

Line No. 16: Prints `Buy an ice-cream` to the console

Line No. 17: Subtracts `icecream_price` from the `money`

Line No. 18: Prints the new value of `money`

Line No. 19: Jumps to **Line No. 14**.

Line No. 21: Prints `You can't buy an ice-cream` to the console

Q

If number = 1, then what is the output of the following code?

```
while (number <= 10) {
    number = number + 1;
  }
  cout << "Number = " << number;
```

Retake Quiz

This sums up our discussion of the `while` loop. Let's discuss the `do-while` loop in the upcoming lesson.