Contents

[Introduction to Loops 1](#_Toc56962992)

[While Loop Demo 1](#_Toc56962993)

[While Loop Demo 1](#_Toc56962994)

[Guess Number 2](#_Toc56962995)

[While Loop 3](#_Toc56962996)

[For Loop Demo 3](#_Toc56962997)

[Summary 4](#_Toc56962998)

## Introduction to Loops

A **loop** is a programming tool that *repeats* some code or a set of instructions until a specified condition is reached. As a programmer, you’ll find that you rely on loops all the time! You’ll hear the generic term “iterate” when referring to loops; *iterate* simply means “to repeat”.

When we see that a process has to repeat multiple times in a row, we write a loop. Loops allow us to create efficient code that automates processes to make scalable, manageable programs.

In this lesson, we will learn about two types of loops: while loops and for loops!

# While Loop Demo

So first up… the while loop!

Before we dive deep into the syntax of the while loop, let’s do a demo.

Inside **enter\_pin.cpp**, we have a program that asks and checks for a password. It uses a while loop to ask the user for the password over and over again.

**Note:** You don’t need to understand the code right now.

## 

## While Loop Demo

#include <iostream>

int main() {

  int pin = 0;

  int tries = 0;

  std::cout << "BANK OF IRELAND\n";

  std::cout << "Enter your PIN: ";

  std::cin >> pin;

  while (pin != 1234 && tries <= 3) {

    std::cout << "Enter your PIN: ";

    std::cin >> pin;

    tries++;

  }

  if (pin == 1234) {

    std::cout << "PIN accepted!\n";

    std::cout << "You now have access.\n";

  }

}

## Guess Number

So now that we got a demo of loops, let’s write one!

The while loop looks very similar to an if statement. And just like an if statement, it executes the code inside of it if the condition is true.

However, the difference is that the while loop will continue to execute the code inside of it, *over and over again*, as long as the condition is true.

Here is what a while loop looks like:

while (condition) {

statements

}

In other words, instead of executing *if* something is true, it executes *while* that thing is true.

while (guess != 8) {  
   
  std::cout << "Wrong guess, try again: ";  
  std::cin >> guess;  
   
}

In this example, while guess is not equal to 8, the program will keep on asking the user to input a new number. It will exit the while loop once the user types in 8 and then the program will continue.

## While Loop

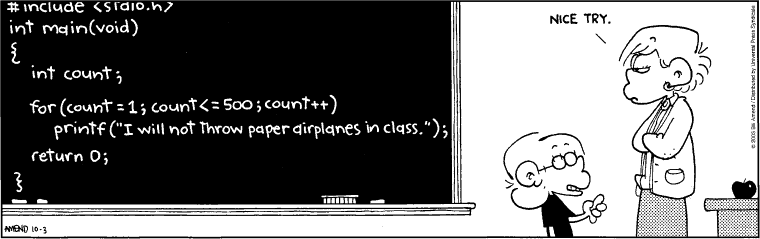
The last one we held your hand, so let’s try one on your own.

As an example of iteration, we have the first program ever to run on a stored-program computer (the [EDSAC](https://en.wikipedia.org/wiki/EDSAC)). It was written and run by David Wheeler in the computer laboratory at Cambridge University, England, on May 6th, 1948, to calculate and print a simple list of squares like the following:

0   0  
1   1  
2   4  
3   9  
4   16  
5   25  
6   36  
7   49  
8   64  
9   81

On the left, there are numbers from 0 to 9. On the right are their squares. For example, for the number 9, 9 \* 9 = 81.

## For Loop Demo



Iterating over a sequence of numbers is so common that C++, like most other programming languages, has a special syntax for it.

When we know exactly how many times we want to iterate (or when we are counting), we can use a for loop instead of a while loop:

for (int i = 0; i < 20; i++)   
{  
   
  std::cout << "I will not throw paper airplanes in class.\n";  
   
}

Let’s take a closer look at the first line:

for (int i = 0; i < 20; i++)

There are three separate parts to this separated by ;:

* The initialization of a counter: int i = 0
* The continue condition: i < 20
* The change in the counter (in this case an increment): i++

So here we are creating a variable i that starts from 0. We will repeat the code inside over and over again when i is less than 20. At the end the for loop, we are adding 1 to i using the ++ operator.

## Summary

* Loops perform repetitive actions so we don’t have to code those actions manually every time.
* How to write while loops with a continue condition.
* How to write for loops with a counter that increments or decrements.