

# Loops

## while Loop

A `while` loop statement repeatedly executes the code block within as long as the condition is `true`. The moment the condition becomes `false`, the program will exit the loop.

Note that the `while` loop might not ever run. If the condition is `false` initially, the code block will be skipped.

```
while (password != 1234) {  
  
    std::cout << "Try again: ";  
    std::cin >> password;  
  
}
```

## for Loop

A `for` loop executes a code block a specific number of times. It has three parts:

- The initialization of a counter
- The continue condition
- The increment/decrement of the counter

```
for (int i = 0; i < 10; i++) {  
  
    std::cout << i << "\n";  
  
}
```

This example prints 0 to 9 on the screen.

"i < 10" is checked before each iteration. "i++" is executed right after each iteration. The declaration "int i = 0" is lo

In C++, there are many different ways of classifying errors, but they can be boil down to four categories: Compile

