Functions in C++

Introduction to Functions

A *function* in C++ contains a set of instructions that are executed when it is called.
A function declaration is composed of three parts:

1. Function return type
2. Function name
3. Function parameters

A function can be called by specifying its name followed by a pair of parentheses `()` .

```
#include <iostream>

void printTitle() {
  std::string msg = "Codecademy\n";
  std::cout << msg;
}

int main(){
  printTitle();

  return 0;
}
```

Function Parameters

When calling a function with multiple parameters, the number and order of the arguments must match with the parameters.
Default parameters initialize to a default value if an argument is not provided in the function call.
Pass by reference lets the function modify the arguments variables. Use the `&` operator to indicate that a parameter is passed by reference.

```
#include <iostream>

double totalPrice(int items, double price
= 9.99) {
  return items * price;
}

// Pass by reference
void addOne(int &i) {
  i += 1;
}

int main() {
  std::cout << totalPrice(10) << "\n";
// Output: 99.9

  int num = 2;
  addOne(num);
  std::cout << num;      // Output: 3

  return 0;
}
```

Function Overloading

With *function overloading*, C++ functions can have the same name but handle different input parameters. At least one of the following criteria must be true in order for functions to be properly overloaded:

- Each function has different types of parameters.
- Each function has a different number of parameters.

The function return type is NOT used to differentiate overloaded functions.

```
#include <iostream>

int add(int a, int b) {
  return a + b;
}

double add(double a, double b) {
  return a + b;
}

int add(int a, int b, int c) {
  return a + b + c;
}

int main() {
  std::cout << add(3, 2); // Calls
add(int, int)
  std::cout << "\n";
  std::cout << add(5.3, 1.4); // Calls
add(double, double)
  std::cout << "\n";
  std::cout << add(2, 6, 9);  // Calls
add(int, int, int)
}
```

Command Line Arguments

Command line arguments are optional arguments passed to the `main()` function of a C++ program. Passing command line arguments is as easy as appending the arguments after the executable name. For example:

```
./greeting Hello World
```

In order to access command line arguments, the new form of `main()` takes two arguments:

- `argc` : the number of command line arguments.
- `argv` : an array containing the values of command line arguments.

```
#include <iostream>

int main(int argc, char* argv[]) {
  std::cout << argc << "\n";

  for(int i = 0; i < argc; i++) {
    std::cout << argv[i] << "\n";
  }
  return 0;
}
```

Print   Share ▾