EC327: Lab 4

Functions

Updated by Miguel Mark, Stephen Tucker, John Mikulskis
mmark9@bu.edu
sstucker@bu.edu
jkulskis@bu.edu
rishavde@bu.edu

Why functions?

Functions are lines of code that are grouped together to perform a task within a scope

Why functions:

- Abstract (Hide) complex operations from the user ie. trig functions
    - User only needs to know arguments, return value, and what the function does!
- Readable code
- Efficient code

Functions in the main program

They go before main so that if you call them, your computer knows what they are.

Format:

```
returnType name(argument)
{
    Code
}

int main()
{
    mainCode
}
```

```
int square(int num)
{
    return num *= num;
}

int main()
{
    int x = square(3);
}
```

What is x?

Created by Steven Maloney; stevenm@bu.edu

Scope

In this block of code we have function ‘square’

- Scope is the ‘box’ within where the variables are usable and visible

What is the value of num at the end of the code?

What is the value of x at the end of the code?

```
int square(int num)
{
    return num *= num;
}

int main()
{
    int num = 3;
    int x = square(num);
}
```

Created by Steven Maloney; stevenm@bu.edu

Organizing functions with files

For complicated projects, it is often helpful to put reused functions into their own files!

By convention, **definitions** are placed in header files .h
Implementations are placed in the .cpp file.

There can be **multiple** .h with **corresponding** .cpp files, if you want
Example file structures:
Function.h
Function2.h
Function.cpp
Function2.cpp

Step 1 - Header file

Header files end in “.h” and should be named the same as the .cpp file
Function.cpp ---> Function.h

Format for Function.h:

```
#ifndef FUNCTION_H //if these functions aren’t defined
#define FUNCTION_H //define these functions
returnType name(arguments); //semicolon needed here
returnType name(arguments);
#endif
```

Step 2 - Function implementation .cpp file

Must include corresponding header at the top and **implement** the function

```
#include “Function.h”

returnType name(arguments)
{

      // Do some stuff maybe
      return returnType

}
```

Step 3 - Using the function in main program

Must include corresponding header in main program

```
#include “Function.h”

// Now you can use myFunction!
```

Using the function in other .cpp files

Same as using it in the main program

```
#include “Function.h”

// Now you can use myFunction in this file!
```

Why bother?

Can improve compilation efficiency

.h file provides a useful reference for users of your function!

Much better readability

Makes the main file much smaller

Easier to debug

Questions?

Lab 4 GitHub

- Accept the invite to join the repository for Lab 4 - https://classroom.github.com/a/6QePoqx6
- Look up the assignment on the repository
- Push ALL files that you have written for the assignment to the repo