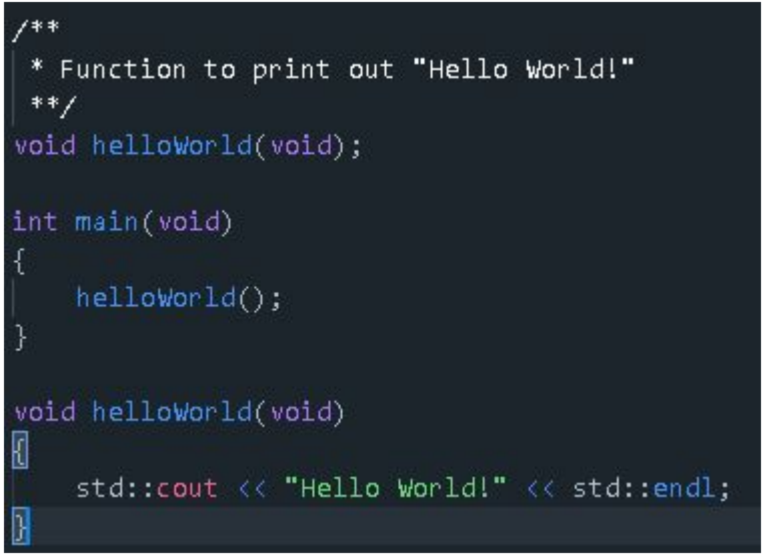
Discussion Section
Week 3

Header Files, Functions, and Includes

Review - Functions

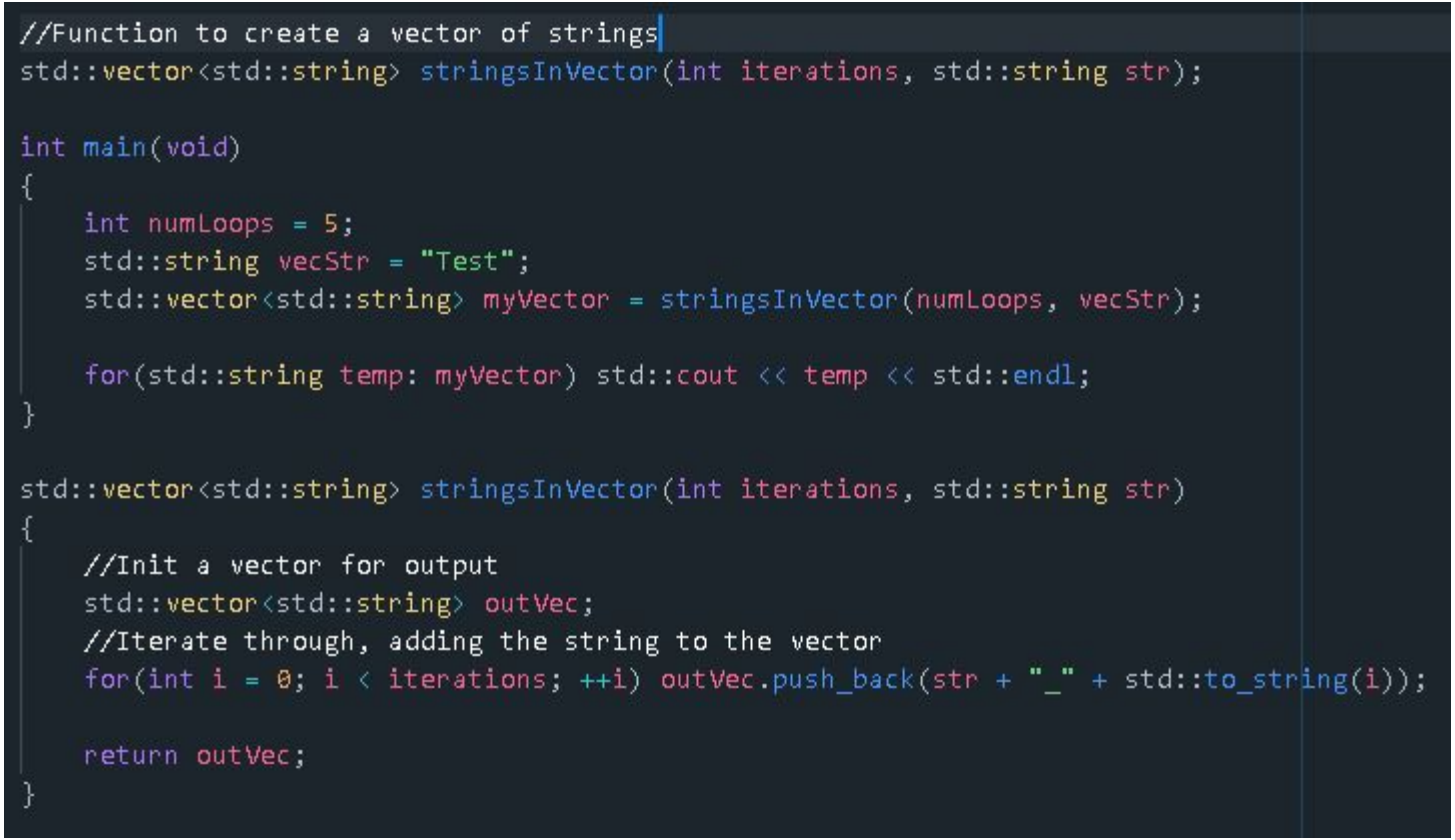
- Functions are blocks of code to perform some action



```
/**
 * Function to print out "Hello World!"
 **/
void helloWorld(void);

int main(void)
{
    helloWorld();
}

void helloWorld(void)
{
    std::cout << "Hello World!" << std::endl;
}
```

Review (cont.)

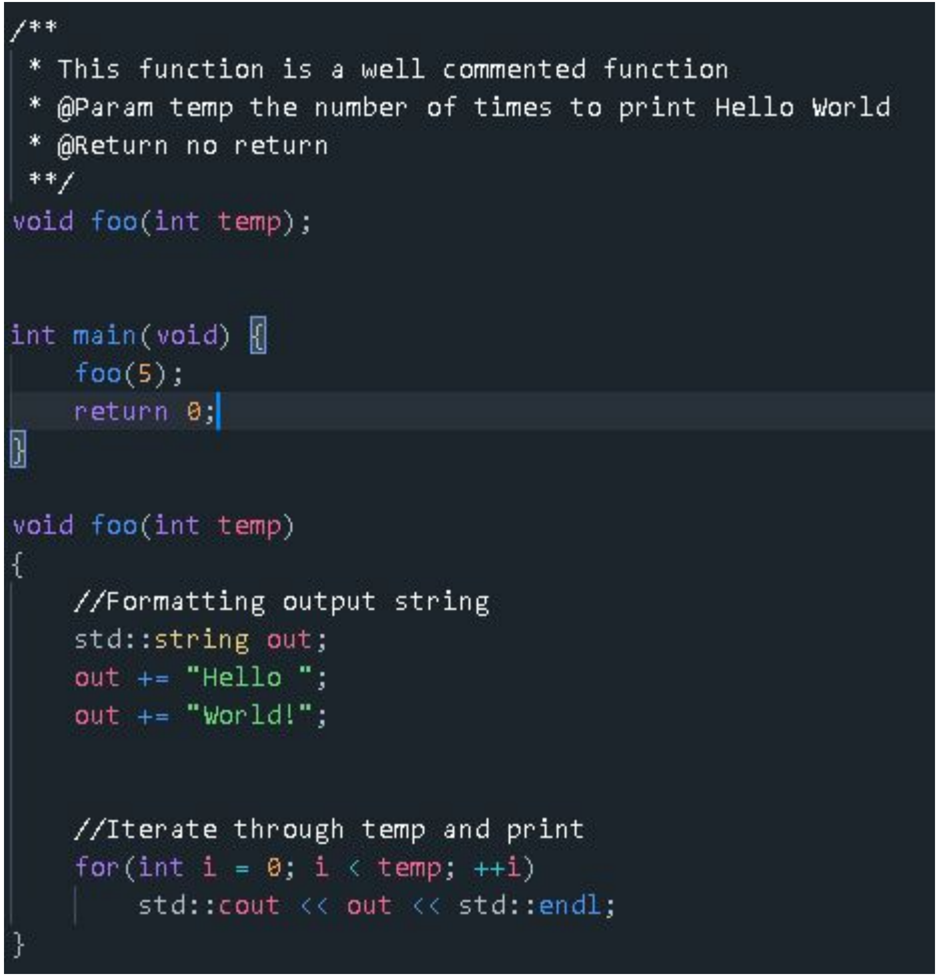
- Parameters
- Types
- Returns
- **Comments**



```
//Function to create a vector of strings
std::vector<std::string> stringsInVector(int iterations, std::string str);

int main(void)
{
    int numLoops = 5;
    std::string vecStr = "Test";
    std::vector<std::string> myVector = stringsInVector(numLoops, vecStr);

    for(std::string temp: myVector) std::cout << temp << std::endl;
}

std::vector<std::string> stringsInVector(int iterations, std::string str)
{
    //Init a vector for output
    std::vector<std::string> outVec;
    //Iterate through, adding the string to the vector
    for(int i = 0; i < iterations; ++i) outVec.push_back(str + "_" + std::to_string(i));

    return outVec;
}
```

Comments

- “Goldilocks”
- JavaDoc Style



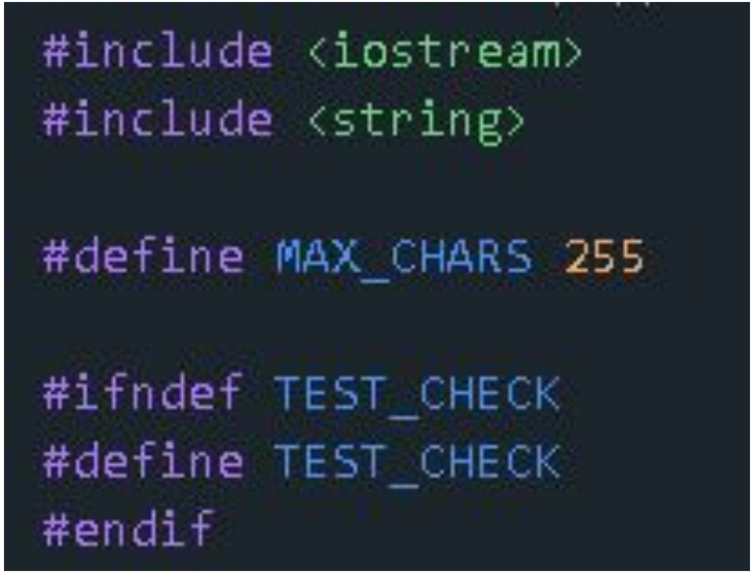
```
/**
 * This function is a well commented function
 * @Param temp the number of times to print Hello World
 * @Return no return
 **/
void foo(int temp);


int main(void) {
    foo(5);
    return 0;
}

void foo(int temp)
{
    //Formatting output string
    std::string out;
    out += "Hello ";
    out += "World!";


    //Iterate through temp and print
    for(int i = 0; i < temp; ++i)
        std::cout << out << std::endl;
}
```

Exercise 1 (5 Minutes)

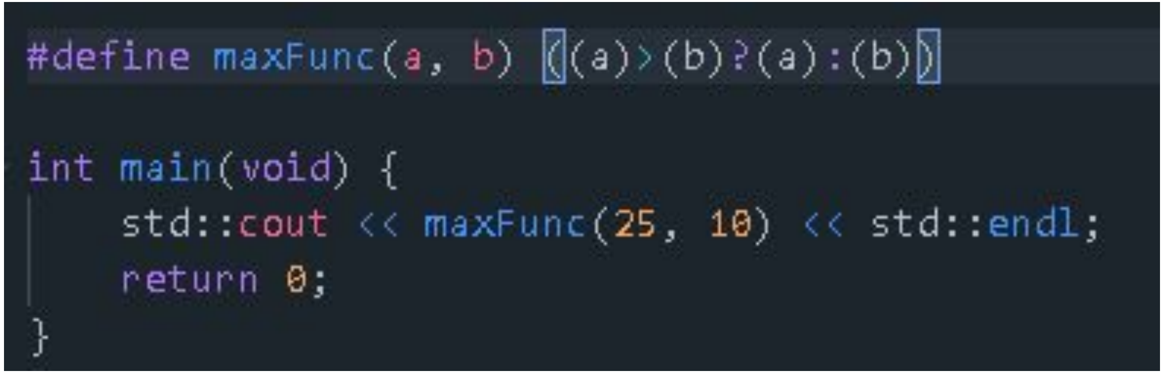
1) Create a “main.cpp” file
    a) Write one function that takes an int, prints out numbers from 0 to that int, and does not return any values
        i) Call the function in main

Include/Preprocessors

- What does it really do?
- Libraries
- Headers



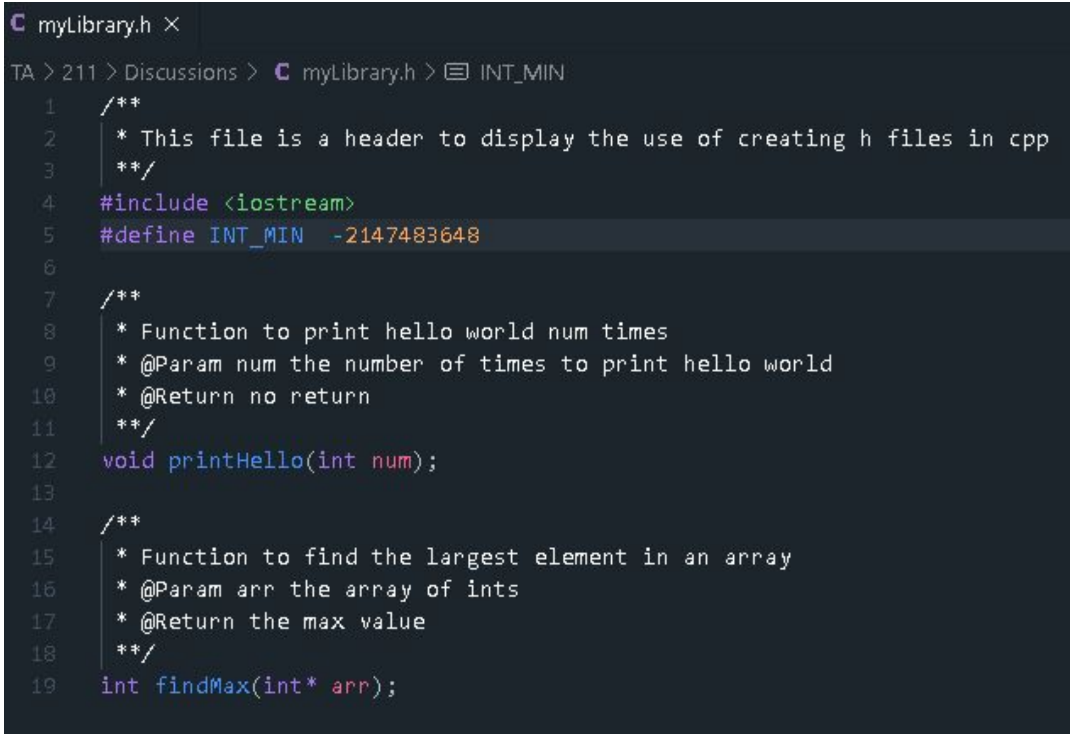
```
#include <iostream>
#include <string>

#define MAX_CHARS 255

#ifndef TEST_CHECK
#define TEST_CHECK
#endif
```



```
#define maxFunc(a, b) ((a)>(b)?(a):(b))

int main(void) {
    std::cout << maxFunc(25, 10) << std::endl;
    return 0;
}
```

Creating Your Own Headers

- What are header files?
- What are they useful for?



```
/**
 * This file is a header to display the use of creating h files in cpp
 **/
#include <iostream>
#define INT_MIN  -2147483648

/**
 * Function to print hello world num times
 * @Param num the number of times to print hello world
 * @Return no return
 **/
void printHello(int num);

/**
 * Function to find the largest element in an array
 * @Param arr the array of ints
 * @Return the max value
 **/
int findMax(int* arr);
```

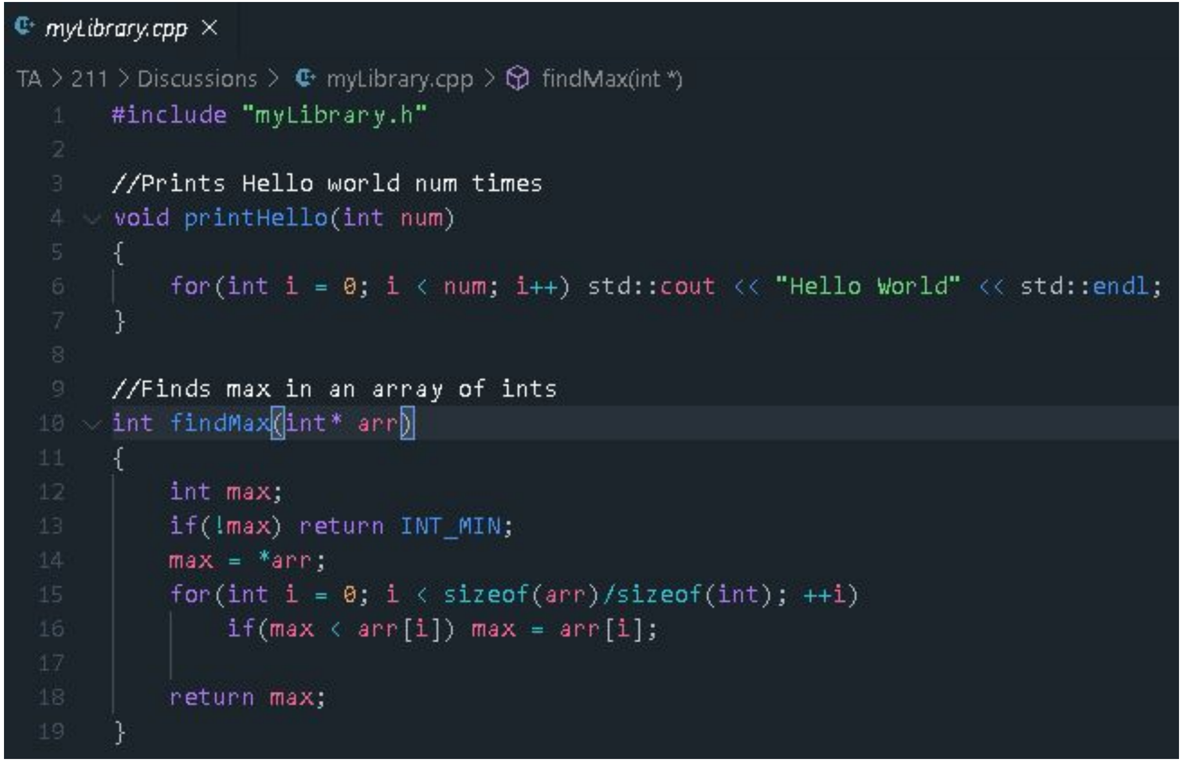
Why do it?

Google

Raytheon Technologies

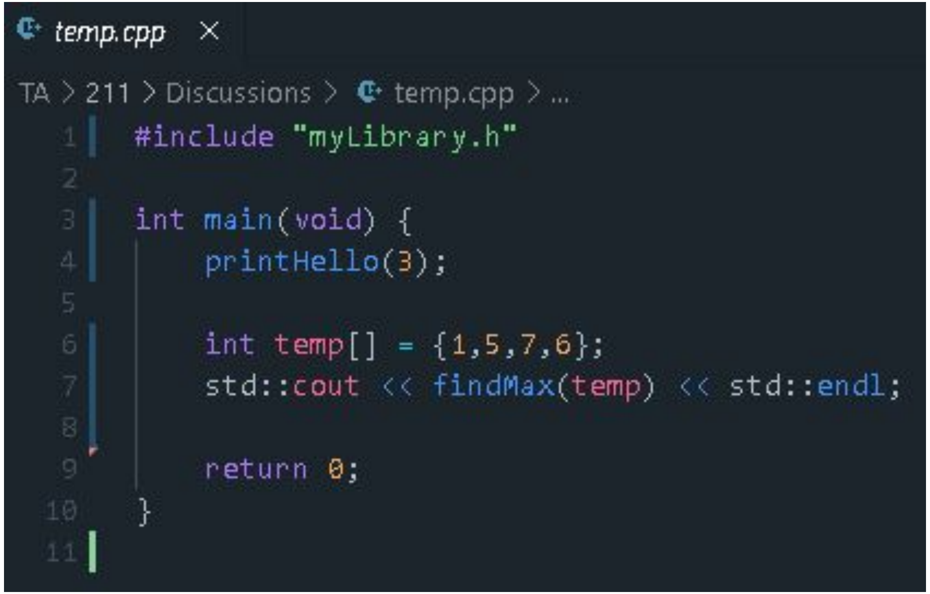
amazon

Headers.....They Don’t Stop With a .h



```cpp
#include "myLibrary.h"

//Prints Hello world num times
void printHello(int num)
{
    for(int i = 0; i < num; i++) std::cout << "Hello World" << std::endl;
}

//Finds max in an array of ints
int findMax(int* arr)
{
    int max;
    if(!max) return INT_MIN;
    max = *arr;
    for(int i = 0; i < sizeof(arr)/sizeof(int); ++i)
        if(max < arr[i]) max = arr[i];

    return max;
}
```

Headers...(cont)

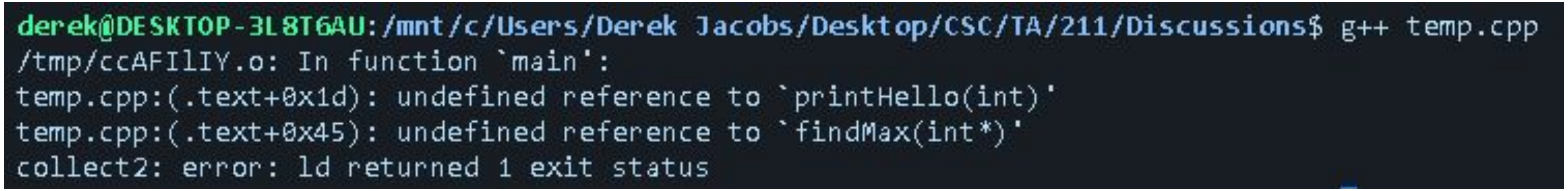


```cpp
#include "myLibrary.h"

int main(void) {
    printHello(3);

    int temp[] = {1,5,7,6};
    std::cout << findMax(temp) << std::endl;

    return 0;
}
```

A note on headers

- Anything you include in your header file will carry over to where it’s included
  - Remember “Copy and Paste”
  - Includes are best left in headers, makes for cleaner code

How to make this compile



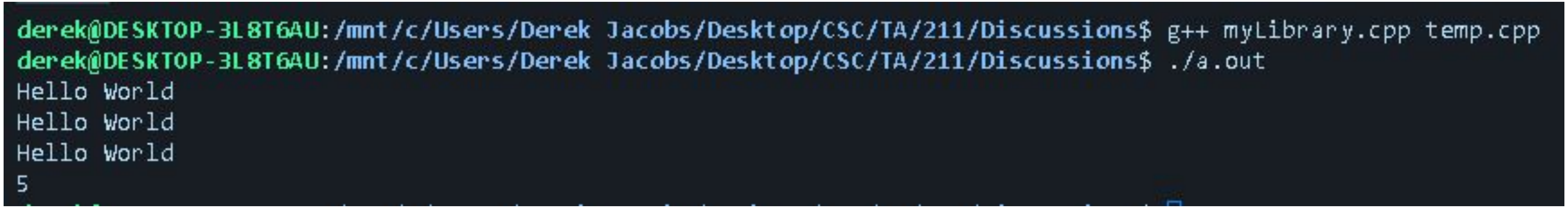
```
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211/Discussions$ g++ temp.cpp
/tmp/ccAFIlIY.o: In function `main':
temp.cpp:(.text+0x1d): undefined reference to `printHello(int)'
temp.cpp:(.text+0x45): undefined reference to `findMax(int*)'
collect2: error: ld returned 1 exit status
```

How to make this compile?

- “Short and easy”
- Make a library!

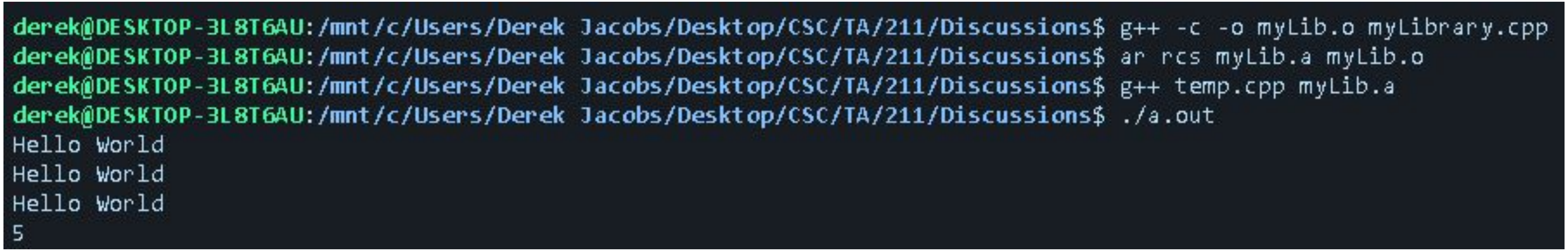
Short and Easy Method

g++ {libraryName.cpp} {targetFile.cpp}



```
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211/Discussions$ g++ myLibrary.cpp temp.cpp
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211/Discussions$ ./a.out
Hello World
Hello World
Hello World
5
```

Making a Library



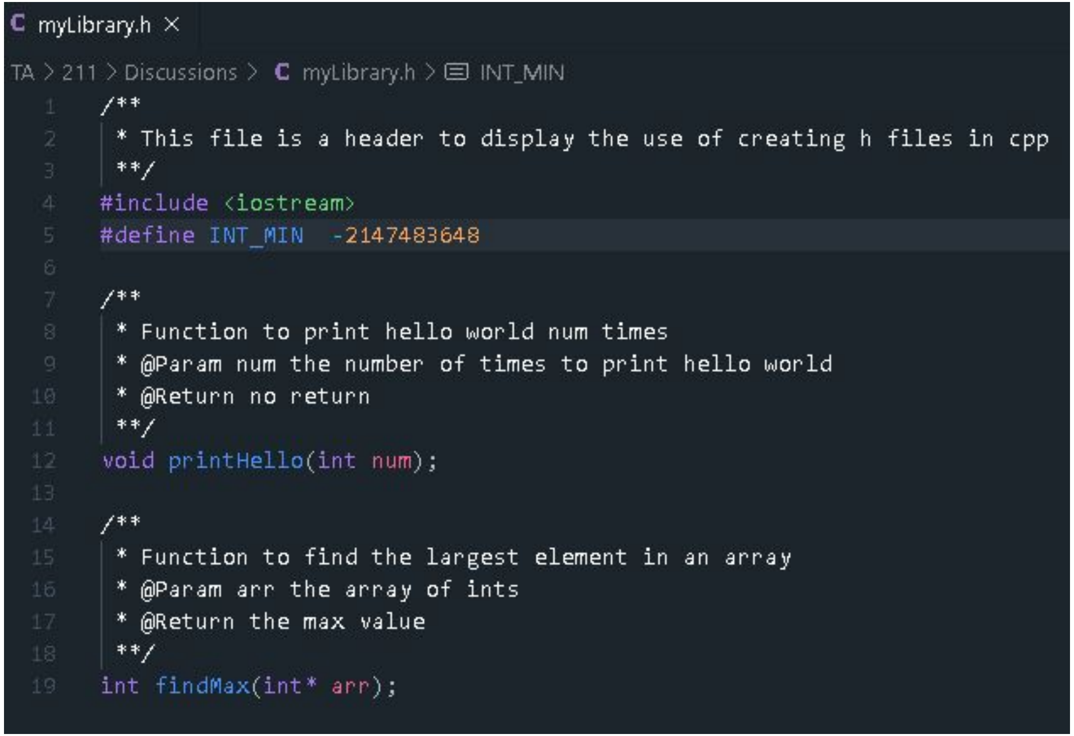
```
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211/Discussions$ g++ -c -o myLib.o myLibrary.cpp
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211/Discussions$ ar rcs myLib.a myLib.o
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211/Discussions$ g++ temp.cpp myLib.a
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211/Discussions$ ./a.out
Hello World
Hello World
Hello World
5
```

When to use each

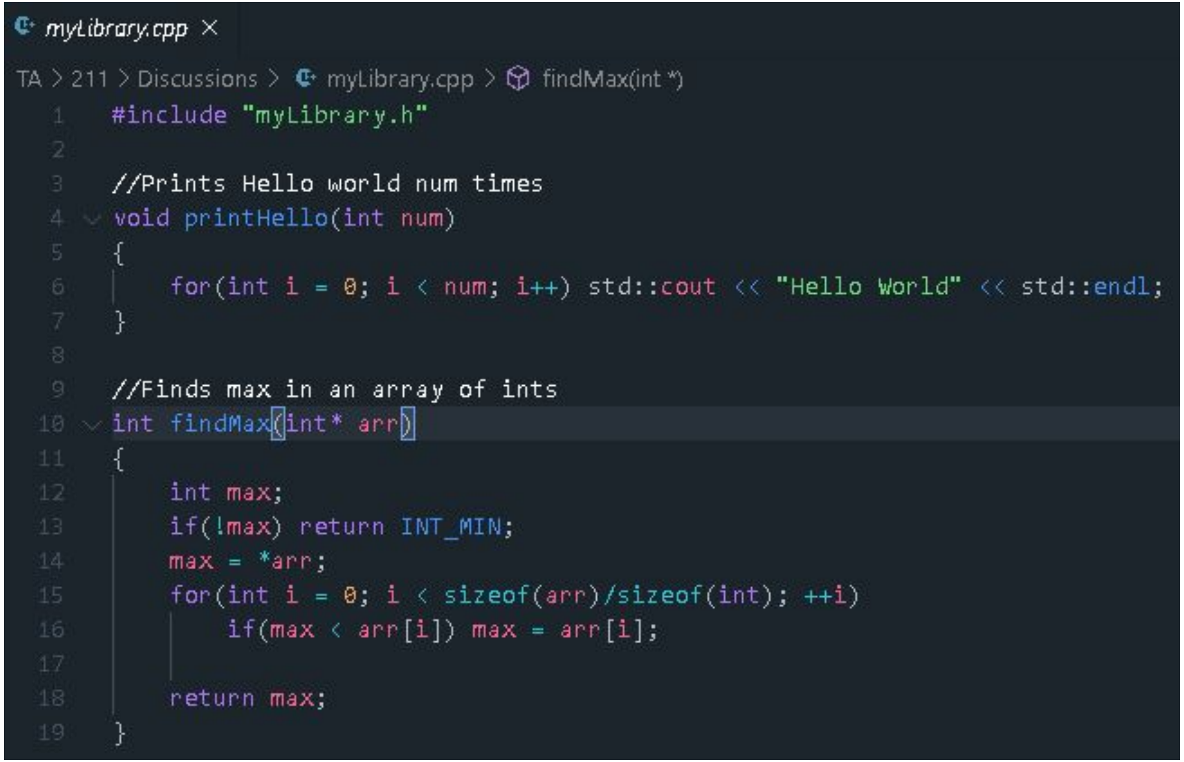
Short and Easy	**Making a Library**
- Small library to be made - Library is not used by many different files	- Large library that is used widely - Prevent recompilation - Much more common in industry

Comparing: Libraries vs. In-Main Declarations and Definitions

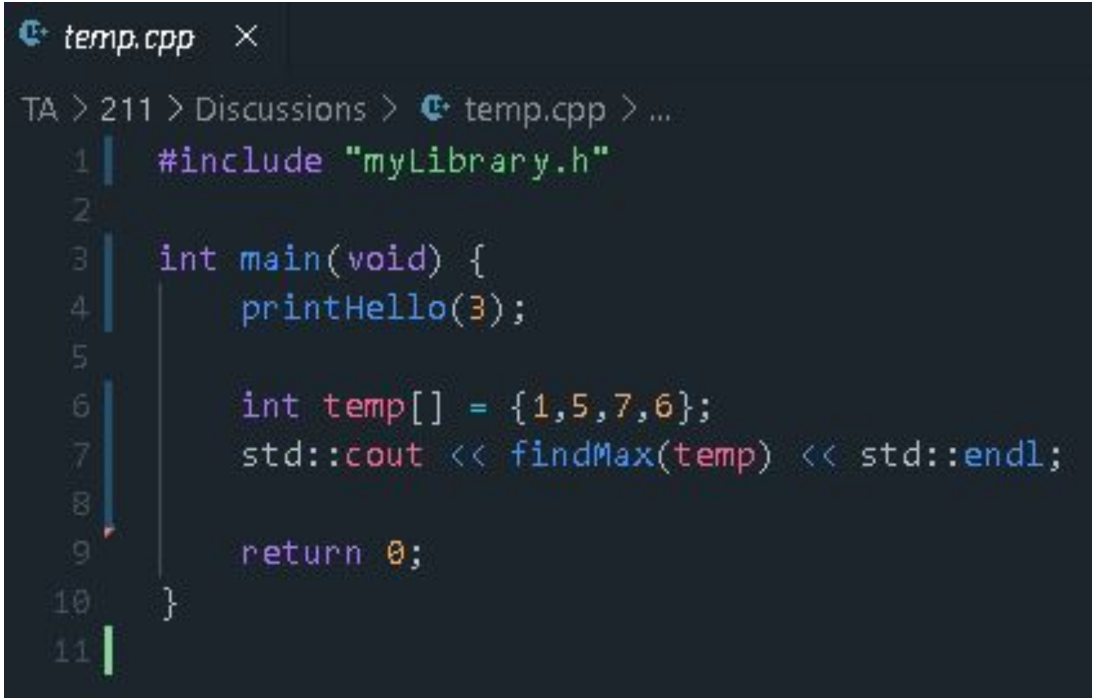
Recall...



```
/**
 * This file is a header to display the use of creating h files in cpp
 **/
#include <iostream>
#define INT_MIN  -2147483648

/**
 * Function to print hello world num times
 * @Param num the number of times to print hello world
 * @Return no return
 **/
void printHello(int num);

/**
 * Function to find the largest element in an array
 * @Param arr the array of ints
 * @Return the max value
 **/
int findMax(int* arr);
```

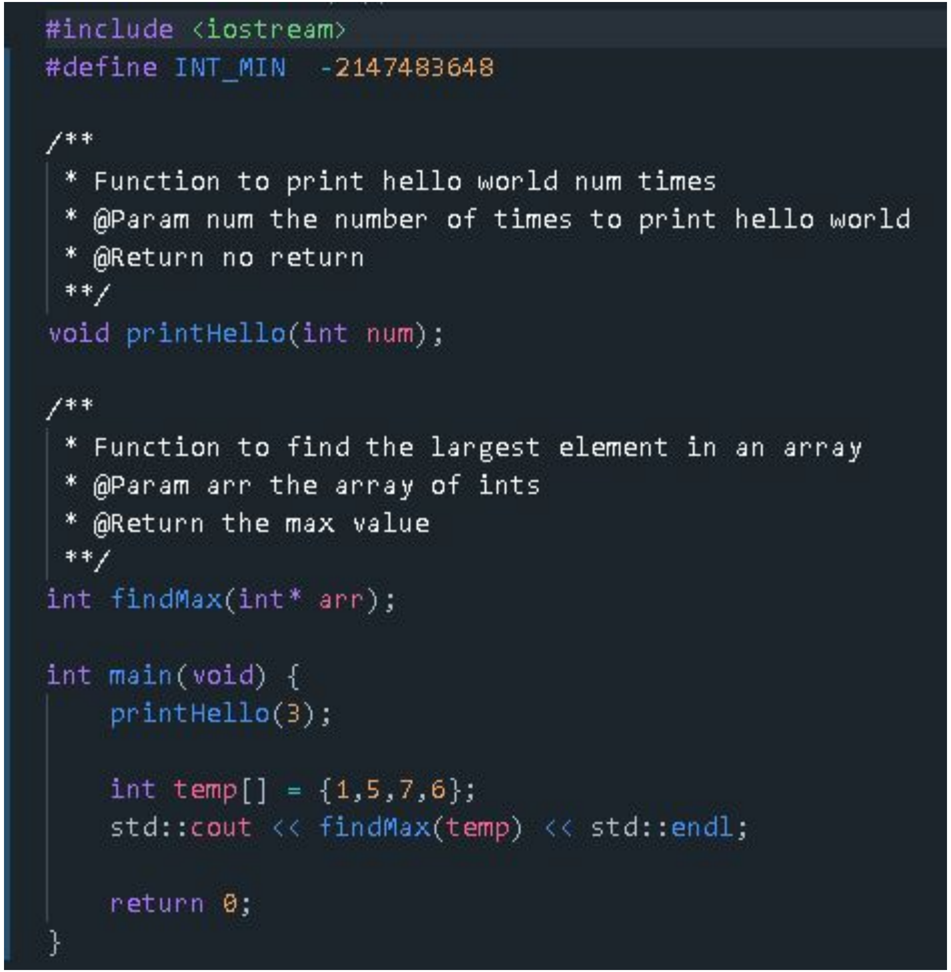


```cpp
#include "myLibrary.h"

//Prints Hello world num times
void printHello(int num)
{
    for(int i = 0; i < num; i++) std::cout << "Hello World" << std::endl;
}

//Finds max in an array of ints
int findMax(int* arr)
{
    int max;
    if(!max) return INT_MIN;
    max = *arr;
    for(int i = 0; i < sizeof(arr)/sizeof(int); ++i)
        if(max < arr[i]) max = arr[i];

    return max;
}
```

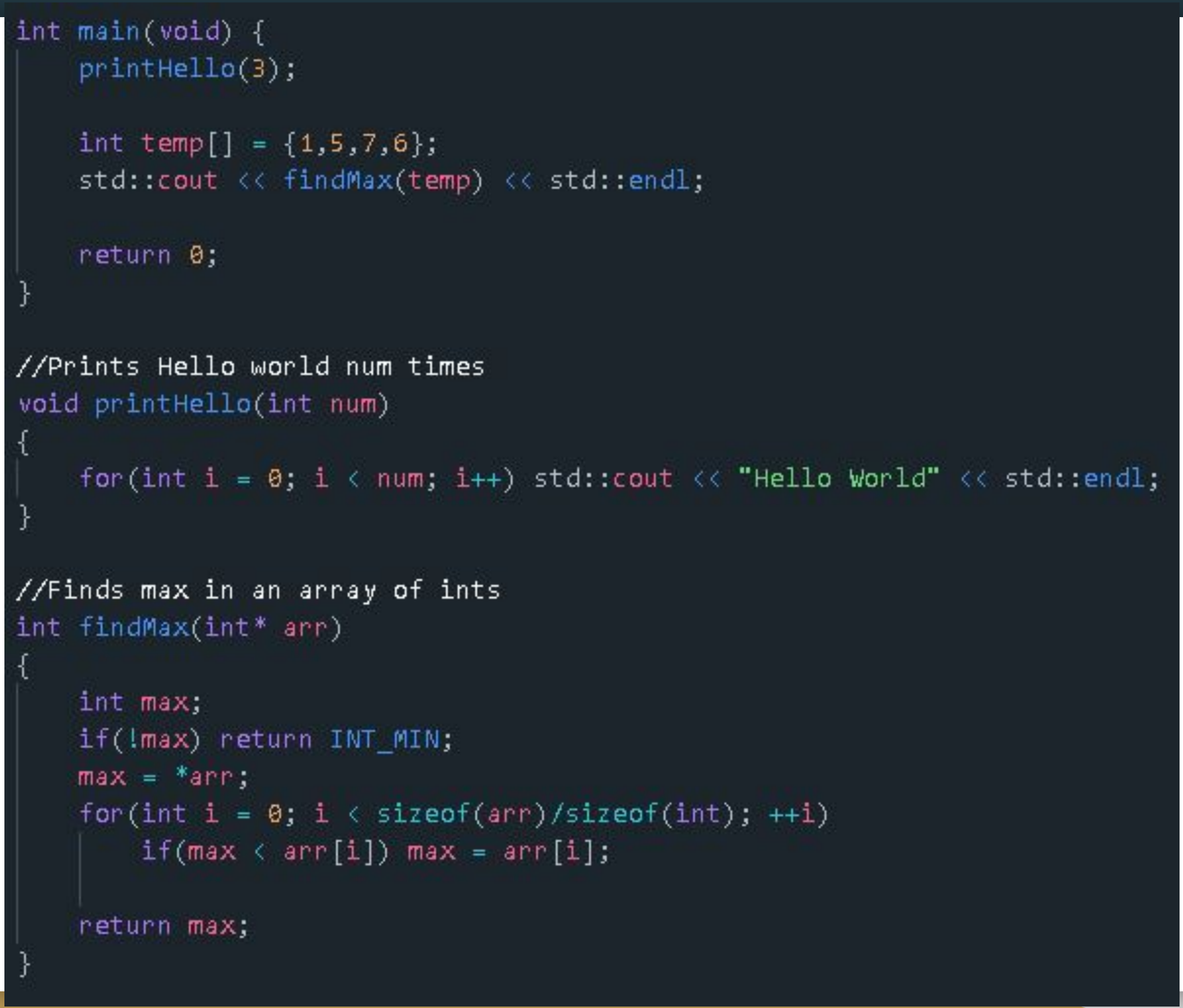


```cpp
#include "myLibrary.h"

int main(void) {
    printHello(3);

    int temp[] = {1,5,7,6};
    std::cout << findMax(temp) << std::endl;

    return 0;
}

```

The same code in one file



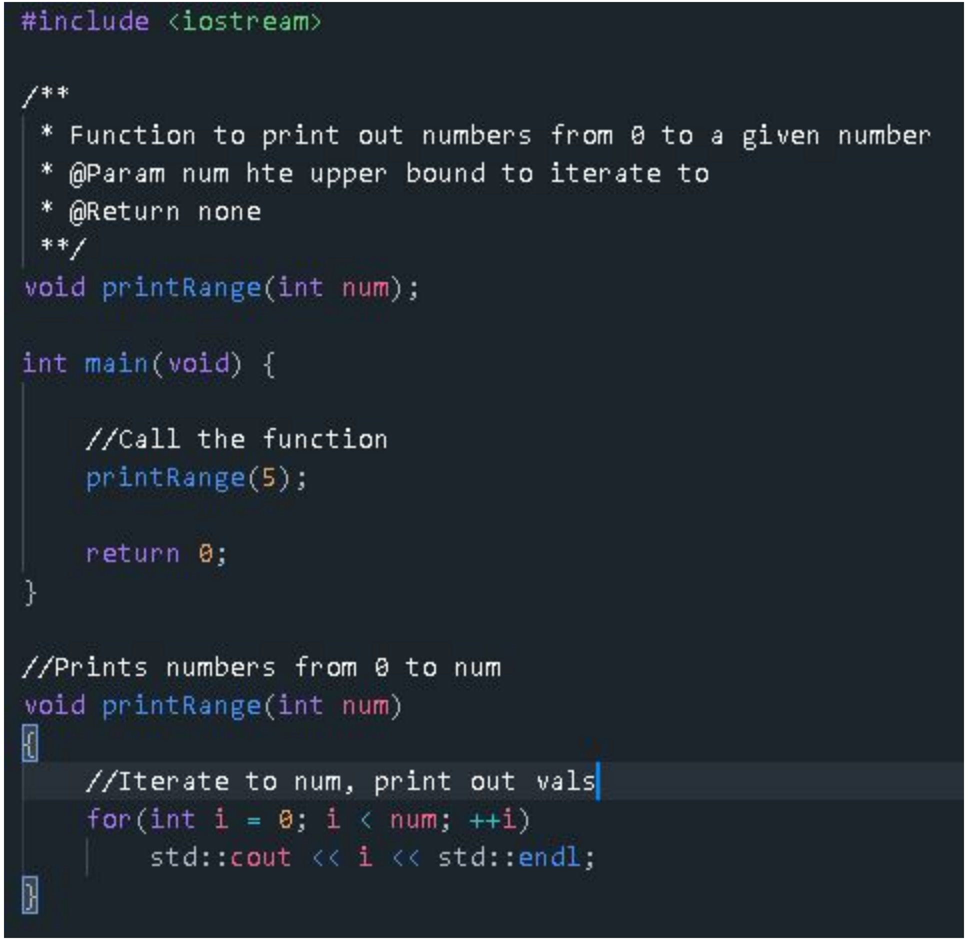
```
#include <iostream>
#define INT_MIN  -2147483648

/**
 * Function to print hello world num times
 * @Param num the number of times to print hello world
 * @Return no return
 **/
void printHello(int num);

/**
 * Function to find the largest element in an array
 * @Param arr the array of ints
 * @Return the max value
 **/
int findMax(int* arr);

int main(void) {
    printHello(3);

    int temp[] = {1,5,7,6};
    std::cout << findMax(temp) << std::endl;

    return 0;
}
```



```
int main(void) {
    printHello(3);

    int temp[] = {1,5,7,6};
    std::cout << findMax(temp) << std::endl;

    return 0;
}

//Prints Hello world num times
void printHello(int num)
{
    for(int i = 0; i < num; i++) std::cout << "Hello World" << std::endl;
}

//Finds max in an array of ints
int findMax(int* arr)
{
    int max;
    if(!max) return INT_MIN;
    max = *arr;
    for(int i = 0; i < sizeof(arr)/sizeof(int); ++i)
        if(max < arr[i]) max = arr[i];

    return max;
}
```

Best Practice:

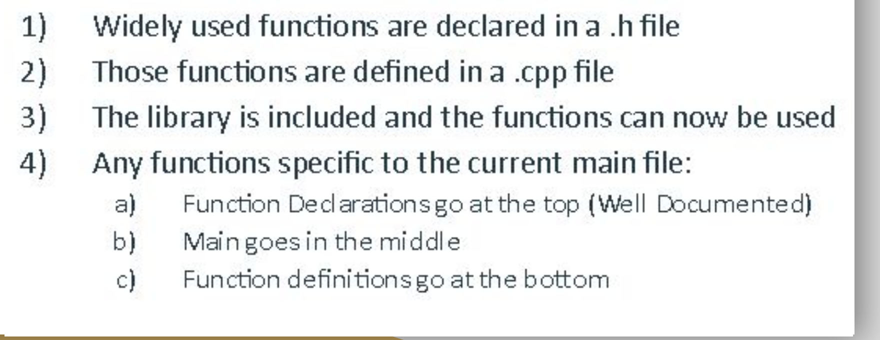
1) Widely used functions are declared in a .h file
2) Those functions are defined in a .cpp file
3) The library is included and the functions can now be used
4) Any functions specific to the current main file:
    a) Function Declarations go at the top (Well Documented)
    b) Main goes in the middle
    c) Function definitions go at the bottom



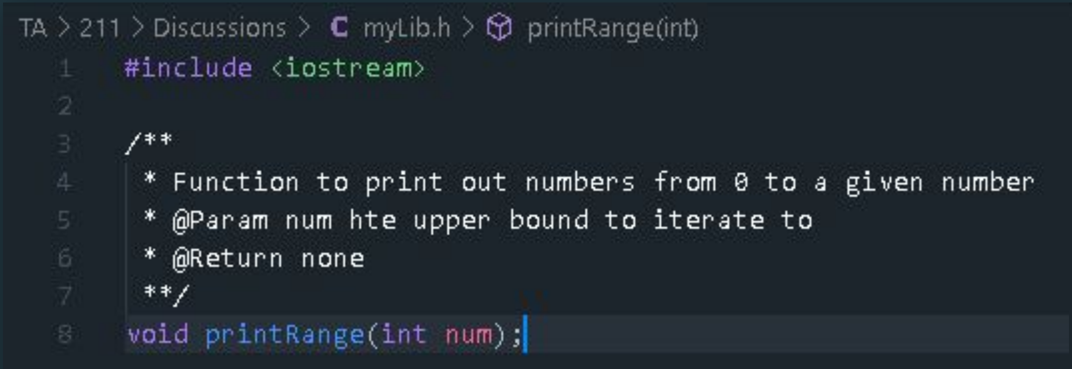
```cpp
#include <iostream>

/**
 * Function to print out numbers from 0 to a given number
 * @Param num hte upper bound to iterate to
 * @Return none
 **/
void printRange(int num);

int main(void) {

    //Call the function
    printRange(5);

    return 0;
}

//Prints numbers from 0 to num
void printRange(int num)
{
    //Iterate to num, print out vals
    for(int i = 0; i < num; ++i)
        std::cout << i << std::endl;
}
```

Exercise 2 (10 Minutes)

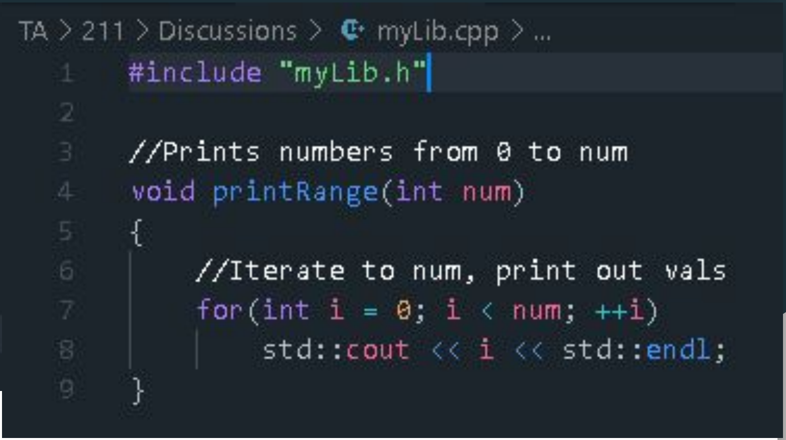
1) Convert your “printRange” function into a library
   a) Again using best practices
   b) Caveat: You can only include iostream once
2) Compile your main file using “The easy way”, and run it, all on one terminal line



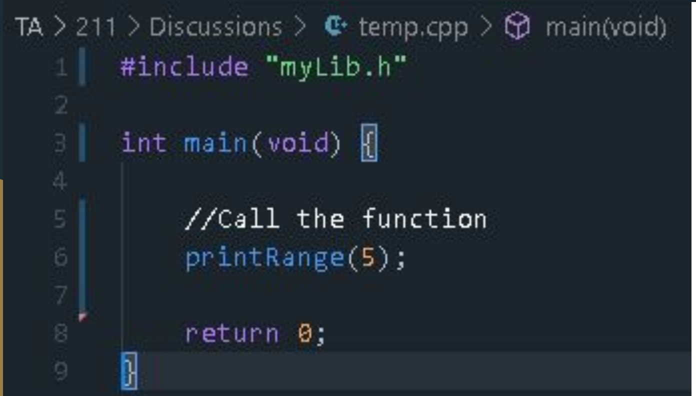
1) Widely used functions are declared in a .h file
2) Those functions are defined in a .cpp file
3) The library is included and the functions can now be used
4) Any functions specific to the current main file:
   a) Function Declarations go at the top (Well Documented)
   b) Main goes in the middle
   c) Function definitions go at the bottom



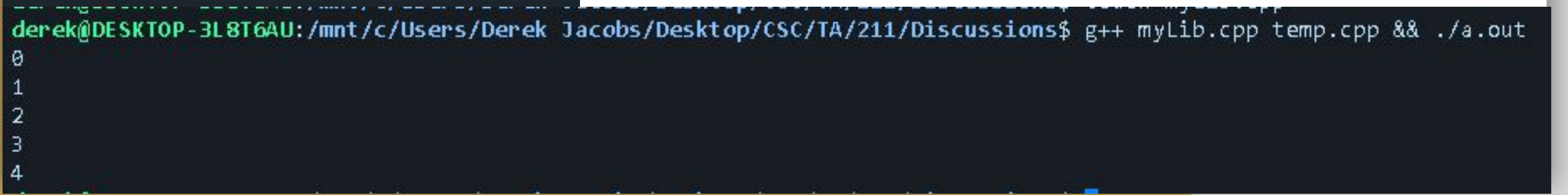
```cpp
#include <iostream>

/**
 * Function to print out numbers from 0 to a given number
 * @Param num hte upper bound to iterate to
 * @Return none
 **/
void printRange(int num);
```



```cpp
#include "myLib.h"

//Prints numbers from 0 to num
void printRange(int num)
{
    //Iterate to num, print out vals
    for(int i = 0; i < num; ++i)
        std::cout << i << std::endl;
}
```



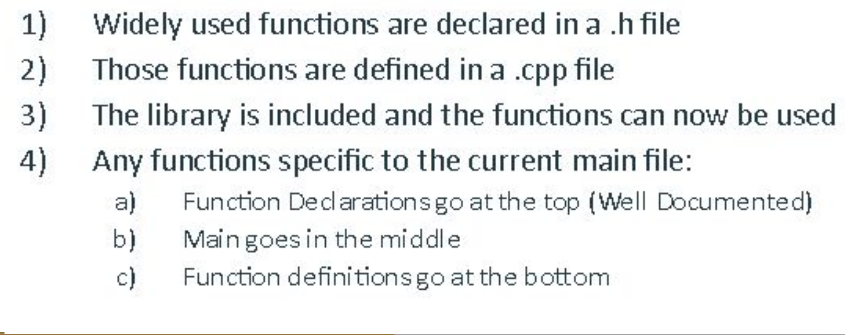
```cpp
#include "myLib.h"

int main(void) {

    //Call the function
    printRange(5);

    return 0;
}
```



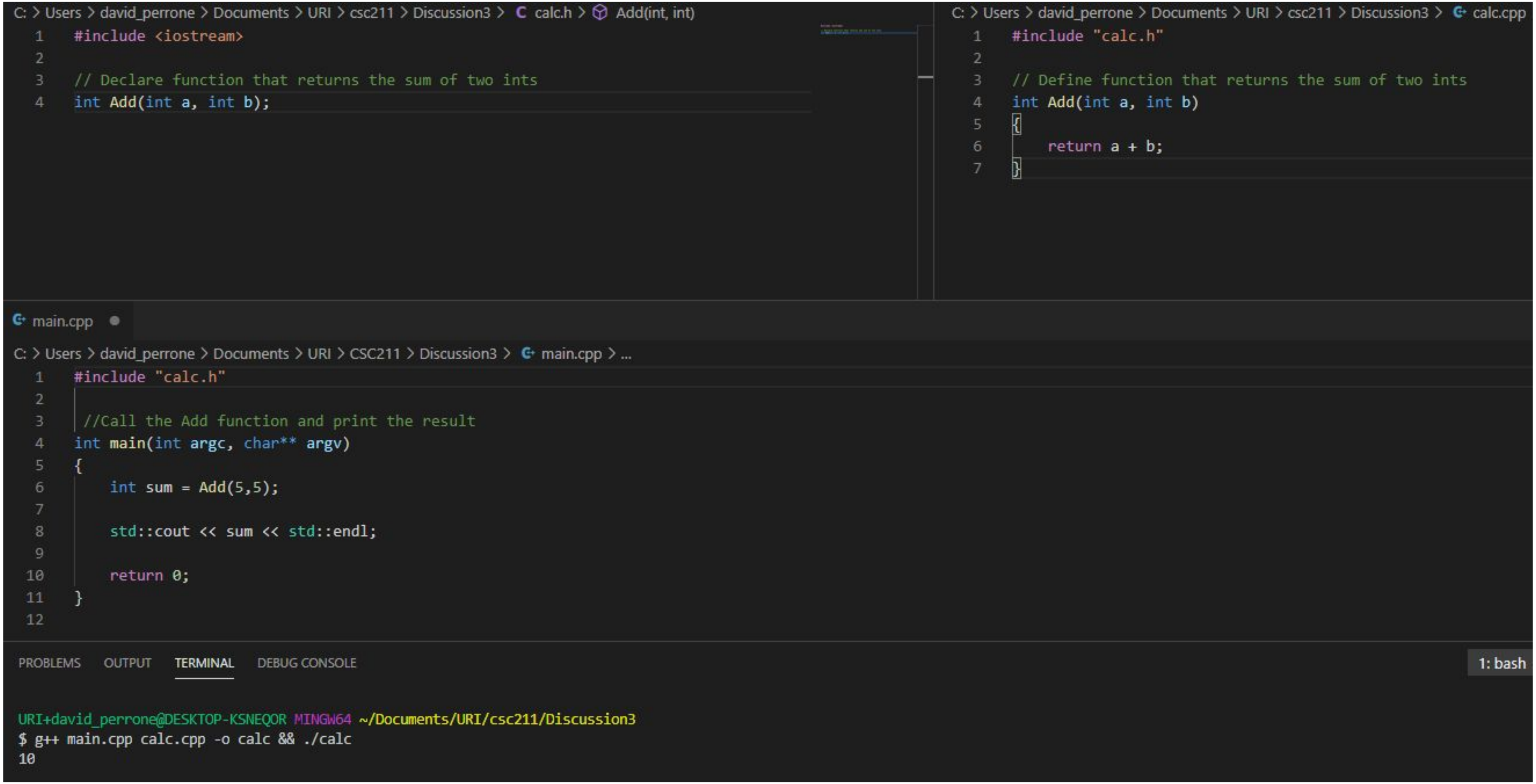
```
derek@DESKTOP-3L8T6AU:/mnt/c/Users/Derek Jacobs/Desktop/CSC/TA/211/Discussions$ g++ myLib.cpp temp.cpp && ./a.out
0
1
2
3
4
```

Exercise 3 (10 Minutes)

1) Create a main.cpp, calc.cpp, and calc.h file
   a) Again using best practices
   b) Caveat: You can only include iostream once
2) Declare a function called “Add” in your calc.h file and define it in your calc.cpp
   a) Takes in two ints as parameters and returns the sum of those ints
   b) In main, call your “Add” function and print the results
3) Compile and run your code in one terminal line



1) Widely used functions are declared in a .h file
2) Those functions are defined in a .cpp file
3) The library is included and the functions can now be used
4) Any functions specific to the current main file:
   a) Function Declarations go at the top (Well Documented)
   b) Main goes in the middle
   c) Function definitions go at the bottom


C: > Users > david_perrone > Documents > URI > csc211 > Discussion3 > calc.h > Add(int, int)
C: > Users > david_perrone > Documents > URI > csc211 > Discussion3 > calc.cpp
main.cpp
C: > Users > david_perrone > Documents > URI > CSC211 > Discussion3 > main.cpp > ...
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
1: bash

```cpp
#include <iostream>

// Declare function that returns the sum of two ints
int Add(int a, int b);
```

```cpp
#include "calc.h"

// Define function that returns the sum of two ints
int Add(int a, int b)
{
    return a + b;
}
```

```cpp
#include "calc.h"

//Call the Add function and print the result
int main(int argc, char** argv)
{
    int sum = Add(5,5);

    std::cout << sum << std::endl;

    return 0;
}

```

```
URI+david_perrone@DESKTOP-KSNEQOR MINGW64 ~/Documents/URI/csc211/Discussion3
$ g++ main.cpp calc.cpp -o calc && ./calc
10
```

Getting Credit

Please email me with some screenshots of your code to show you tried