CS 1580 Lab 7: Functions and Multiple Files

Objective: To work with Functions using multiple files and concepts of **Pass by Value** and **Pass by Reference.**

Lab Definition: The goal is to write getValue() and swapFunc() functions which get values and swap the values of two integer variables. You have to implement these using **Pass by Reference.**

Important: Create multiple files for this program that should implement the objective for this lab. ***Add Programmer name, Student ID, Section, File, description in all the files.***

Note:

Use **fg++ *.cpp –o out1** to compile multiple CPP files.

Sample Output:

Enter the value for num1: 100

Enter the value for num2: 200

Before swapping, the value of num1: 100 num2: 200

After swapping, the value of num1: 200 num2: 100.

Steps:

1. Make a new directory named Lab7 under cs1580 folder and go into that directory
   - cd SDRIVE/cs1580/
   - mkdir lab7
   - cd lab7
2. Open three new files:
   - (jpico “lab_7”.cpp)
   - (jpico “functions_file_name”.cpp)
   - (jpico “header_name”.h)

3. Write and Compile your code: (fg++ *.cpp - o out1)

4. Run your program: (./out1)
5. Submit the program using command

- 4:00 p.m. to 5:50 p.m. →**cssubmit 1580 f 7**
- 6:00 p.m. to 8:50 p.m. →**cssubmit 1580 g 7**

Your program will be graded on:

- Use of pass-by-value and pass-by-reference appropriately.
- Use of multiple files.
- Use of meaningful variable names – they will not be x and y.
- Appropriate indentations/ commenting / Header Comments.
- Use of functions and write pre, post, and description for each prototype.
- Use return even in void functions.
- Readability and Correctness of the program.

Hints:

- Use a temporary variable to do the swap() within the swap function.
- The getValue() will not have any separate variable declarations, just *cout* and *cin* statements.
- You will not be using pointers in this lab.

Function prototype sample:

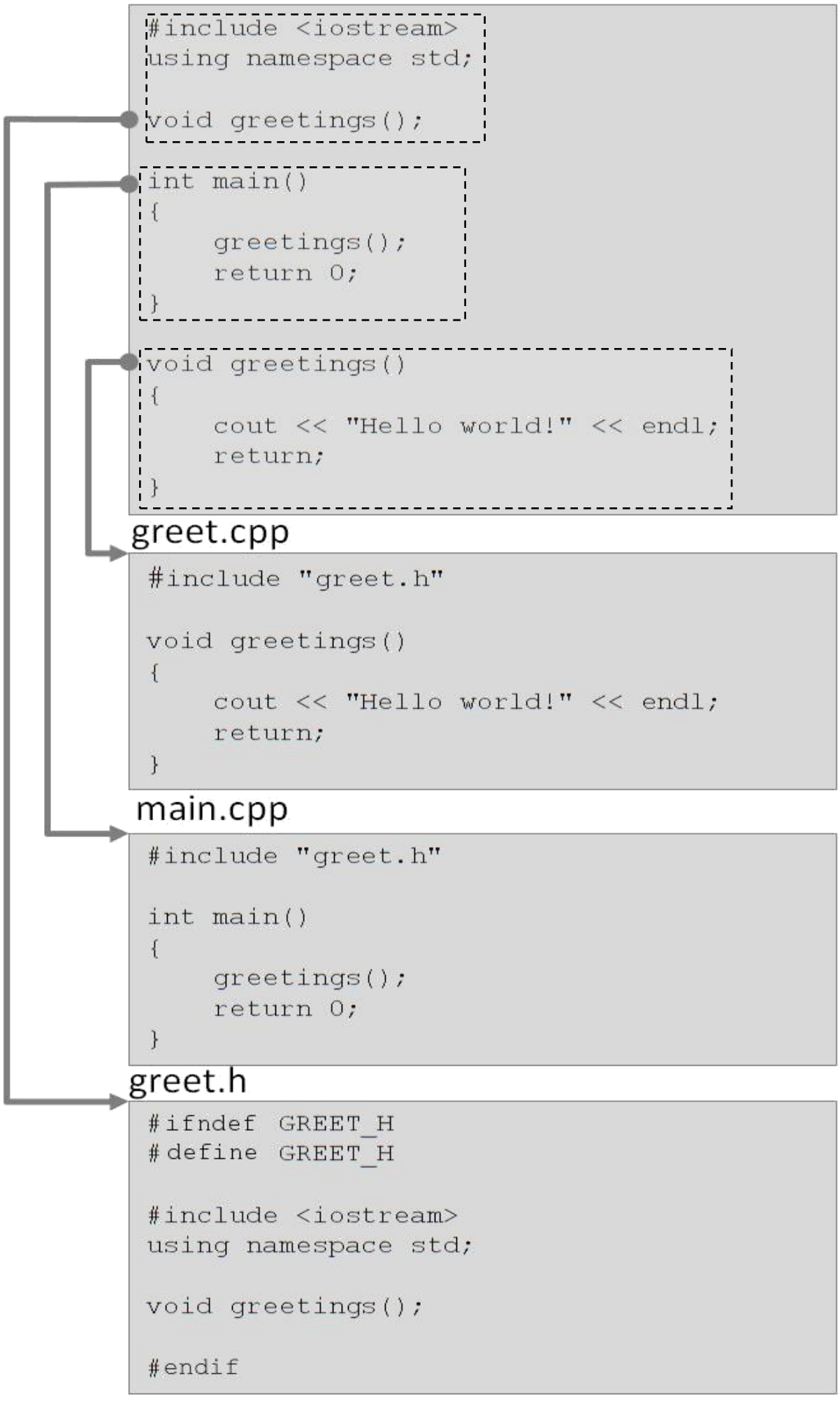
```
float function_name (int &var1, int &var2);
```

Function definition sample:

```
float function_name (int &var1, int &var2)
{
var1 = value1;
var2 = value2;
}
```

Function Call sample:

```
function_name (var1, var2);
```


#include <iostream>
using namespace std;
void greetings();
int main()
{
greetings();
return 0;
}
void greetings()
{
cout << "Hello world!" << endl;
return;
}
greet.cpp
#include "greet.h"
void greetings()
{
cout << "Hello world!" << endl;
return;
}
main.cpp
#include "greet.h"
int main()
{
greetings();
return 0;
}
greet.h
#ifndef GREET_H
#define GREET_H
#include <iostream>
using namespace std;
void greetings();
#endif