CMPD 244

Programming II

Lab 2 – Part 1: Function in C++

In this lesson, you are supposed to learn on how to write and use your own (user defined) function in C++.

You must also know the difference between passing argument by address reference, or value of a variable reference. There are 3 main components to enable user defined function:

1. Function Prototype.

2. Function Definition

3. Function Call

**Program Sample 1:** The following codes will show examples of user defined function.

#include <iostream>

using namespace std;

void draw\_circle (); // Draw a circle.

void draw\_triangle (); // Draw a triangle.

void draw\_intersect (); // Draw intersecting lines.

void draw\_base ();

void main ()

{

// Draw the figure.

draw\_circle (); // Draw a circle.

draw\_triangle (); // Draw a triangle.

draw\_intersect (); // Draw intersecting lines.

}

// DRAWS A CIRCLE

void draw\_circle ()

{

cout << " \* " << endl;

cout << " \* \*" << endl;

cout << " \* \* " << endl;

} // end draw\_circle

// DRAWS A TRIANGLE

void draw\_triangle ()

{

// Draw a triangle.

draw\_intersect ();

draw\_base ();

} // end draw\_triangle

// DRAWS INTERSECTING LINES

void draw\_intersect ()

{

cout << " / \\ " << endl;

cout << " / \\ " << endl;

cout << " / \\" << endl;

} // end draw\_intersect

// DRAWS A HORIZONTAL LINE

void draw\_base ()

{

cout << " -------" << endl;

} // end draw\_base

**Program Sample 2:** Function with Parameters by value

#include <iostream>

using namespace std;

int maximum( int, int, int ); // function prototype

int main()

{

int a, b, c;

cout << "Enter three integers: " << endl;

cin >> a >> b >> c; // a, b and c below are arguments to the maximum function call

cout << "Maximum is: " << maximum( a, b, c ) << endl;

//return 0;

}

// Function maximum definition

// x, y and z below are parameters to

// the maximum function definition

int maximum( int x, int y, int z )

{

int max = x;

if ( y > max )

max = y;

if ( z > max )

max = z;

return max;

}

**Lab Exercises 1:**

1. Write a program in function to compute the square of a number.
2. Write TWO (2) functions to cater for TWO (2) different messages that will be displayed to user based on their annual income. For instance, if your user’s annual income is more that RM40,000 per year, then a congratulatory message will be displayed to them, or else an encouragement message will be displayed if their annual income is less than RM40,000 per year.

Lab 2 – Part 2: Modular Programming in C++

In this lesson, you are supposed to learn on how to implement modular programming in C++. In the concept of modular programming, the C++ codes (and often function prototypes) are normally split up into two files. The header file has the extension of **(.h)** and contains codes definitions and functions. The implementation of the codes goes into the **(.cpp)** file. By doing this, if your codes implementation doesn’t change then it won’t need to be recompiled. Most IDE’s will do this for you – they will only recompile the codes that have changed. This is possible when they are split up this way, but it isn’t possible if everything is in one file (or if the implementation is all part of the header file).

In order for you to implement modular programming in C++, you need to initialize the following steps:-

1. Create a header file with **(.h)** extension. Example: **file\_header\_name.h**

2. Create a source/implementation file with **(.cpp)** extension. Example: **filename.cpp**

To begin, let us consider the following example in Program Sample 1:

**Program Sample 1:**

This program is created via a function named getRandomNumber() to generate the random number and display it to user. Every time this program is executed, user will be able to view various random numbers appearing on their monitor screen. The implementation codes are as below:

//Reuse of functions

//RandomStuff

#include <iostream>

#include <ctime>

#include <cstdlib>

using namespace std;

int getRandomNumber();

int main()

{

int theNumber = getRandomNumber();

cout << theNumber << endl;

return 0;

}

//user-defined functions

int getRandomNumber()

{

srand(time(0));

return rand() % 100+1;

}

Now, let’s re-create this program out so that it will have to separate files; the header file and the implementation file and observe the output. Follow these steps:-

1. First, create a source file with **(.cpp)** extension.

**Go to:** File > New > File... > Visual C++ > C++ File (filename.cpp).

If you want to create a program you should create a project first:

File > New > Project. Then you can add files: Project > Add New Item...

here’s the example….

**RandomStuff.cpp**

//Reuse of functions

//RandomStuff

#include "RandomNumber.h"

using namespace std;

int main()

{

int theNumber = getRandomNumber();

cout << "Your random number now is = " << theNumber << endl;

return 0;

}

1. Second, create a header file the same way you create a source file (.cpp) but this time around, make sure you add a header file to a project;

**Go to:** Project > Add New Item... > Header File (file\_header\_name.h).  
Then you can #include "yourHeader.h" in other files in your project

here’s the example….

**RandomNumber.h**

#include <iostream>

#include <ctime>

#include <cstdlib>

//user-defined functions

int getRandomNumber()

{

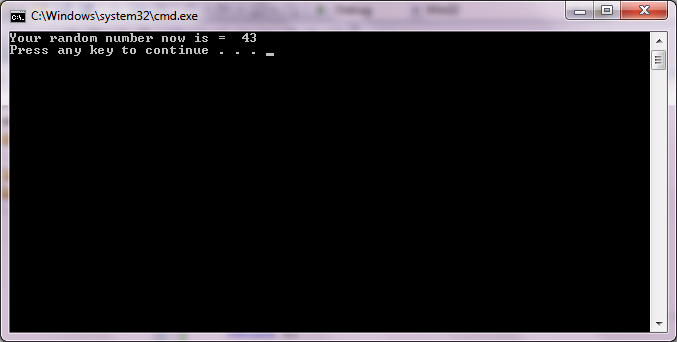
srand(time(0));

return rand() % 100+1;

}

1. Once executed, you should be able to see that the following codes from these 2 files are actually generating the same output just as Program Sample 1 above.

**Sample Output:**

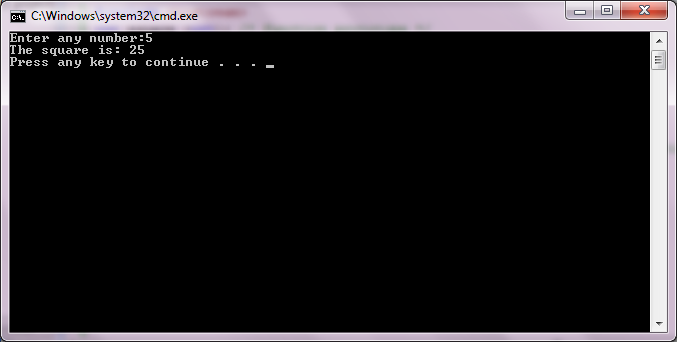


Now that you’re done executing the steps given, you can proceed to experience our lab exercise for today! Voila! ☺

**Lab Exercise 2:**

Modify **Question 1** from the previous **lab exercise 1** (on creating a program in function to compute the square of a number). You need to create 2 separate files, one for header file (.h) and another one for source file (.cpp). You need to ensure that your program will be able to produce the same output as previous program once it is executed.

**Sample Output:**



*Don’t forget to submit the screen capture of your output to our Brighten portal with the subject: CMPD244 Lab2 <YourStudentID><Your Full Name> at the end of this lab session. Good Luck! ☺*

Good Luck! ☺