

SWE20004 Technical Software Development

Lab 5 (week 5)

You will need:

[Online C++ IDE / C++](#)

[Computer installed IDE](#)

A computer with internet access

In this lab you will investigate C++ Functions & References

Before you start the lab exercise, download an appropriate C++ IDE to run your commands and programs.

1. Answer the following questions? Don't copy and paste – write down what it is - in your own words briefly.

- 1.1 When a function calls itself, either directly or indirectly, this is defined as _____

When a function calls itself, either directly or indirectly but has an exit condition, this is defined as recursion.

- 1.2 A function can have _____ parameters

A function can have either zero, many or default parameters.

- 1.3 Variables defined in the body of a function that are visible only to the function are called _____ variables.

Variables defined in the body of a function that are visible only to the function are called local variables.

- 1.4 Variables defined in functions whose values persist from call to call are called _____ variables.

Variables defined in functions whose values persist from call to call are called static variables. This type of variable is only initialized once, and unless initialized, it is automatically initialized to 0.

- 1.5 The default manner in which passing parameters to functions is achieved in C++ _____.

The default manner in which passing parameters to functions is achieved in C++ pass by value. The method is passing arguments to a function that takes the actual value of an argument into the formal parameter of the function. The changes made to the parameter inside the function will not have any effect on the argument.

2. Find the error in the following programs and write the program without any errors in the below box. Please write the program using functions (along with other main () program)

Every changes will be highlighted in **red** (including the other main() program)

a)

```
int g()
{
cout << "Inside function g" << endl;
int h()
{
cout << "Inside function h" << endl;
}
}
```

```
#include <iostream>
using namespace std;
int g()
{
    cout << "Inside function g" << endl;
    return 0;
}
int h()
{
    cout << "Inside function h" << endl;
    return 0;
}

int main()
{
    g();
    h();
    return 0;
}
```

```
1  #include <iostream>
2  using namespace std;
3  int g()
4  {
5      cout << "Inside function g" << endl;
6      return 0;
7  }
8  int h()
9  {
10     cout << "Inside function h" << endl;
11     return 0;
12 }
13
14 int main()
15 {
16     g();
17     h();
18     return 0;
19 }
20
```

```
Inside function g
Inside function h

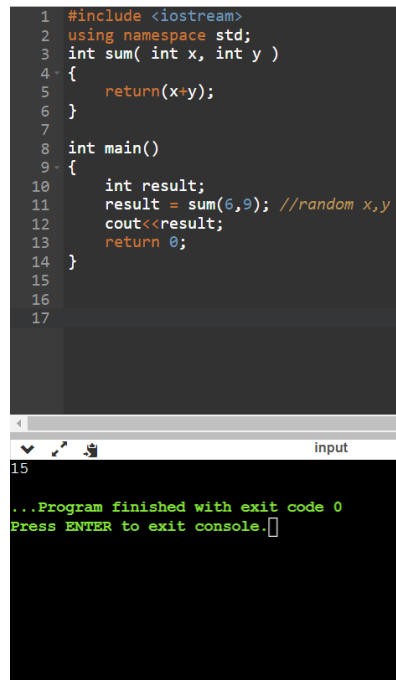
...Program finished with exit code 0
Press ENTER to exit console.
```

b)

```
int sum( int x, int y )
{
    int result;
    result = x + y;
}
```

```
#include <iostream>
using namespace std;
int sum( int x, int y )
{
    return(x+y);
}

int main()
{
    int result;
    result = sum(6,9); //random x,y
    cout<<result;
    return 0;
}
```



```
1 #include <iostream>
2 using namespace std;
3 int sum( int x, int y )
4 {
5     return(x+y);
6 }
7
8 int main()
9 {
10     int result;
11     result = sum(6,9); //random x,y
12     cout<<result;
13     return 0;
14 }
15
16
17
```

input

15

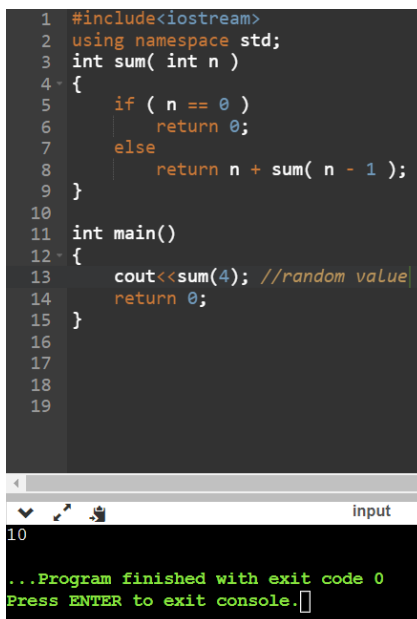
...Program finished with exit code 0
Press ENTER to exit console.

c)

```
int sum( int n )
{
    if ( n == 0 )
        return 0; else
        n + sum( n - 1 );
}
```

```
#include<iostream>
using namespace std;
int sum( int n )
{
    if ( n == 0 )
        return 0;
    else
        return n + sum( n - 1 );
}

int main()
{
    cout<<sum(4); //random value
    return 0;
}
```



```
1 #include<iostream>
2 using namespace std;
3 int sum( int n )
4 {
5     if ( n == 0 )
6         return 0;
7     else
8         return n + sum( n - 1 );
9 }
10
11 int main()
12 {
13     cout<<sum(4); //random value
14     return 0;
15 }
16
17
18
19
```

input

10

...Program finished with exit code 0
Press ENTER to exit console.

d)

```
void f( double a );
{
    float a;
    cout << a << endl;
}
```

```
#include<iostream>
using namespace std;
void f( double a )
{
    cout << a << endl;
}
int main()
{
    f(6.9); //random value
    return 0;
}
```



```
1 #include<iostream>
2 using namespace std;
3 void f( double a )
4 {
5     cout << a << endl;
6 }
7 int main()
8 {
9     f(6.9); //random value
10    return 0;
11 }
12
13
14
```

input

6.9

...Program finished with exit code 0
Press ENTER to exit console.

e)

```
void product()
{
    int a;
    int b; int c;
    int result;
    cout << "Enter three integers: "; cin >> a >> b >> c;

    result = a * b * c;
    cout << "Result is " << result; return result;
}
```

```
#include<iostream>
using namespace std;

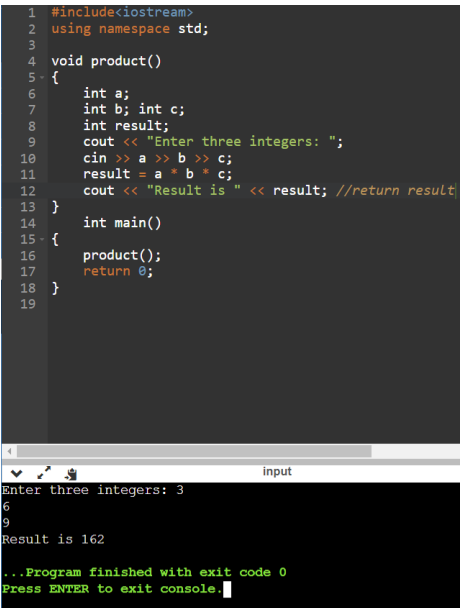
void product()
{
    int a;
    int b; int c;
```

```

int result;
cout << "Enter three integers: ";
cin >> a >> b >> c;
result = a * b * c;
cout << "Result is " << result; //return result
}

int main()
{
    product();
    return 0;
}

```



```

1 #include<iostream>
2 using namespace std;
3
4 void product()
5 {
6     int a;
7     int b; int c;
8     int result;
9     cout << "Enter three integers: ";
10    cin >> a >> b >> c;
11    result = a * b * c;
12    cout << "Result is " << result; //return result
13 }
14 int main()
15 {
16     product();
17     return 0;
18 }
19

```

input

Enter three integers: 3
6
9
Result is 162
...Program finished with exit code 0
Press ENTER to exit console.

f)

```

template < class A >
int sum( int num1, int num2, int num3 )
{
    return num1 + num2 + num3;
}

```

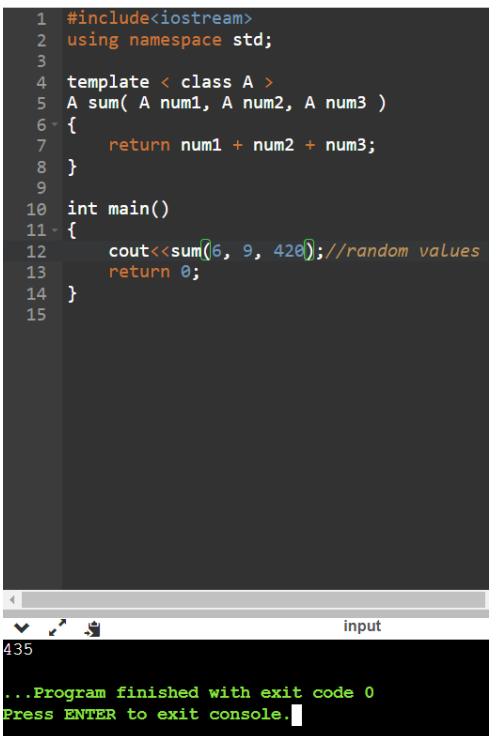
```

#include<iostream>
using namespace std;

template < class A >
A sum( A num1, A num2, A num3 )
{
    return num1 + num2 + num3;
}

int main()
{
    cout<<sum(6, 9, 420); //random values
    return 0;
}

```



```

1 #include<iostream>
2 using namespace std;
3
4 template < class A >
5 A sum( A num1, A num2, A num3 )
6 {
7     return num1 + num2 + num3;
8 }
9
10 int main()
11 {
12     cout<<sum(6, 9, 420); //random values
13     return 0;
14 }
15

```

input

435
...Program finished with exit code 0
Press ENTER to exit console.

g)

```
void printResults( int x, int y )
{
    cout << "The sum is " << x + y << '\n';
    return x + y;
}
```

```
#include<iostream>
using namespace std;
```

```
void printResults( int x, int y )
{
    cout << "The sum is " << x + y << '\n';
}
```

```
int main()
{
    printResults(6, 9); //random values
    return 0;
}
```

```
1 #include<iostream>
2 using namespace std;
3
4 void printResults( int x, int y )
5 {
6     cout << "The sum is " << x + y << '\n';
7 }
8
9 int main()
10 {
11     printResults(6, 9); //random values
12     return 0;
13 }
14
```

The sum is 15

...Program finished with exit code 0
Press ENTER to exit console.

h)

```
template < A >
A product( A num1, A num2, A num3 )
{
    return num1 * num2 * num3;
}
```

```
#include<iostream>
using namespace std;
```

```
template < class A >
A product( A num1, A num2, A num3 )
{
    return num1 * num2 * num3;
}
```

```
int main()
{
    cout<<product(2, 4, 6); //random values
}
```

```
1 #include<iostream>
2 using namespace std;
3
4 template < class A >
5 A product( A num1, A num2, A num3 )
6 {
7     return num1 * num2 * num3;
8 }
9
10 int main()
11 {
12     cout<<product(2, 4, 6); //random values
13     return 0;
14 }
```

48

...Program finished with exit code 0
Press ENTER to exit console.

```
return 0;  
}
```

3. Give the function header for each of the following functions:

- a) Function hypotenuse that takes two double-precision floating-point arguments side1 and side2 and returns a double-precision floating-point result.

Ans: double hypotenuse(double side1, double side2)

- b) Function smallest that takes three integers x, y and z, and returns an integer.

Ans: int smallest(int x, int y, int z)

- c) Function instructions that does not receive any arguments and does not return a value. [Note: Such functions are commonly used to display instructions to a user.]

Ans: void instructions()

- d) Function intToDouble that takes an integer argument number and returns a doubleprecision floating-point result.

Ans: double intToDouble(int number)

4. What is displayed by the following code? Write the output in the box? Also, Explain the difference of this question 4 program concept with question 5 program concept?

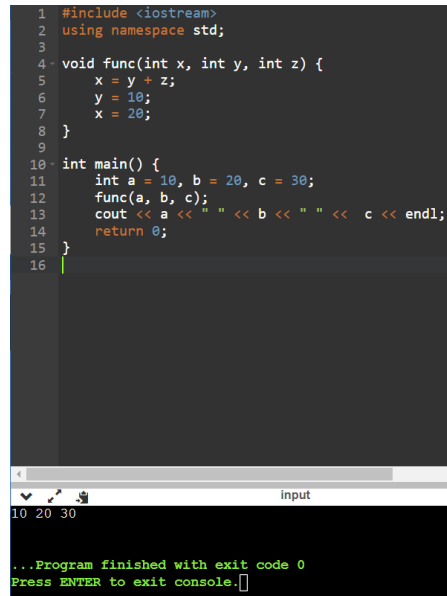
```
. using namespace std;  
.   
. void func(int x, int y, int z) {  
.     x = y + z;  
.     y = 10;  
.     x = 20;  
. }  
.   
. int main() {  
.     int a = 10, b = 20, c = 30;  
.     func(a, b, c);  
.     cout << a << " " << b << " " << c << endl;  
.     return 0;  
. }
```

The changes are highlighted in red. Comparing with question 5, question 5 used the ampersand symbol &, which is used to accept the address (or reference) to a variable, instead of the value of the variable. This means that instead of printing out the variable values in main(), it will print out the values declared in function (in this case, it means printing out the values of x, y, z instead of the values of a, b, c).

```
#include <iostream>
using namespace std;
```

```
void func(int x, int y, int z) {
    x = y + z;
    y = 10;
    x = 20;
}
```

```
int main() {
    int a = 10, b = 20, c = 30;
    func(a, b, c);
    cout << a << " " << b << " " << c << endl;
    return 0;
}
```



```
1 #include <iostream>
2 using namespace std;
3
4 void func(int x, int y, int z) {
5     x = y + z;
6     y = 10;
7     x = 20;
8 }
9
10 int main() {
11     int a = 10, b = 20, c = 30;
12     func(a, b, c);
13     cout << a << " " << b << " " << c << endl;
14     return 0;
15 }
16
```

10 20 30

...Program finished with exit code 0
Press ENTER to exit console.

5. What is displayed by the following code? Write the output in the box?

```
. #include <iostream>
. using namespace std;
.
. void func(int &x, int &y, int &z) {
.     x = y + z;
.     y = 10;
.     x = 20;
. }
.
. int main() {
.     int a = 10, b = 20, c = 30;
.     func(a, b, c);
.     cout << a << " " << b << " " << c << endl;
.     return 0;
. }
```



```
1  #include <iostream>
2  using namespace std;
3
4  void func(int &x, int &y, int &z) {
5      x = y + z;
6      y = 10;
7      x = 20;
8  }
9
10 int main() {
11     int a = 10, b = 20, c = 30;
12     func(a, b, c);
13     cout << a << " " << b << " " << c << endl;
14     return 0;
15 }
16
```

input

20 10 30

...Program finished with exit code 0
Press ENTER to exit console.

6. Write the program using Functions and prototypes - Converting temperatures

In this exercise you will create a program that will be used to convert Fahrenheit temperatures to Celsius and Kelvin temperatures through the use of two functions.

For this program, assume that **temperature will be represented as a double value**. Begin by defining the function prototypes for the functions **fahrenheit_to_celsius** and **Fahrenheit_to_kelvin** which are both passed a double value and **return a double value**.

Now, at the bottom of the file, write the full definitions of both functions.

The function **fahrenheit_to_celsius** is **passed a Fahrenheit temperatures** and **returns a rounded Celsius temperature**. You may use the **<cmath>** function **round** in order to round the return value. The formula to convert Fahrenheit to Celsius is **$(5.0/9.0) * (\text{temperature} - 32)$** .

The function `fahrenheit_to_kelvin` is passed a Fahrenheit temperatures and returns a rounded Kelvin temperature. The formula to convert Fahrenheit to Kelvin is $(5.0/9.0) * (\text{temperature} - 32) + 273$.

Now, from the `temperature_conversation` function, declare and initialize the variables `Celsius_temperature` and `kelvin_temperature` by calling the appropriate functions which you have just defined and passing the variable `fahrenheit_temperature`.

```
#include <iostream>
#include <cmath>
using namespace std;

float fahrenheit_to_celsius(float a){
    return((5.0/9.0)*(a - 32)); }

float fahrenheit_to_kelvin(float a){
    return((5.0/9.0)*(a - 32) + 273); }

int main()
{
    float fah, cel, kel;
    cout<<"Input temperature in fahrenheit: ";
    cin>>fah;
    fahrenheit_to_celsius(fah);
    cel = round(fahrenheit_to_celsius(fah));
    cout<<"The temperature in celsius is: "<<cel<<"\n";
    fahrenheit_to_kelvin(fah);
    kel = round(fahrenheit_to_kelvin(fah));
    cout<<"The temperature in kelvin is: "<<kel;
    return 0;
}
```

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 float fahrenheit_to_celsius(float a){
6     return((5.0/9.0)*(a - 32)); }
7
8 float fahrenheit_to_kelvin(float a){
9     return((5.0/9.0)*(a - 32) + 273); }
10
11 int main()
12 {
13     float fah, cel, kel;
14     cout<<"Input temperature in fahrenheit: ";
15     cin>>fah;
16     fahrenheit_to_celsius(fah);
17     cel = round(fahrenheit_to_celsius(fah));
18     cout<<"The temperature in celsius is: "<<cel<<"\n";
19     fahrenheit_to_kelvin(fah);
20     kel = round(fahrenheit_to_kelvin(fah));
21     cout<<"The temperature in kelvin is: "<<kel;
22     return 0;
23 }
24
```

input

```
Input temperature in fahrenheit: 0
The temperature in celsius is: -18
The temperature in kelvin is: 255

...Program finished with exit code 0
Press ENTER to exit console.
```

7. Write a program for calculator that can add, subtract, multiply and divide two Numbers using Functions. Display the output on screen.

The output will look like this.

Please enter the numbers: 10 20

Please enter the operator: +

The sum of 10 and 20 is: 30

Please enter the numbers: 20 5

Please enter the operator: -

The sum of 20 and 5 is: 15

...

```
#include <iostream>
using namespace std;

float addition(float a, float b){
    return(a + b);}
float subtraction(float a, float b){
    return(a - b);}
float multiplication(float a, float b){
    return(a * b);}
float division(float a, float b){
    return(a / b);}

int main()
{
    float num1, num2, sum, subtr, multp, divs;
    string oprt;
    cout<<"Please enter the 1st number: ";cin>>num1;
    cout<<"Please enter the 2nd number: ";cin>>num2;
    cout<<"Please enter the operator: ";cin>>oprt;
    if (oprt == "+"){
        sum = addition(num1, num2);
        cout<<"The sum of "<<num1<<" and "<<num2<<" is: "<<sum;}
    else if (oprt == "-"){
        subtr = subtraction(num1, num2);
        cout<<"The difference of "<<num1<<" and "<<num2<<" is: "<<subtr;}
    else if (oprt == "*"){
        multp = multiplication(num1, num2);
        cout<<"The multiplication of "<<num1<<" and "<<num2<<" is:
"<<multp;}
    else if (oprt == "/"){
        divs = division(num1, num2);
        cout<<"The division of "<<num1<<" and "<<num2<<" is: "<<divs;}
    else
        cout<<"Invalid operator";
    return 0;
}
```

```

1  #include <iostream>
2  using namespace std;
3
4  float addition(float a, float b){
5      return(a + b);}
6  float subtraction(float a, float b){
7      return(a - b);}
8  float multiplication(float a, float b){
9      return(a * b);}
10 float division(float a, float b){
11     return(a / b);}
12
13 int main()
14 {
15     float num1, num2, sum, subtr, multp, divs;
16     string oprt;
17     cout<<"Please enter the 1st number: ";cin>>num1;
18     cout<<"Please enter the 2nd number: ";cin>>num2;
19     cout<<"Please enter the operator: ";cin>>oprt;
20     if (oprt == "+"){
21         sum = addition(num1, num2);
22         cout<<"The sum of "<<num1<<" and "<<num2<<" is: "<<sum;}
23     else if (oprt == "-"){
24         subtr = subtraction(num1, num2);
25         cout<<"The difference of "<<num1<<" and "<<num2<<" is: "<<subtr;}
26     else if (oprt == "*"){
27         multp = multiplication(num1, num2);
28         cout<<"The multiplication of "<<num1<<" and "<<num2<<" is: "<<multp;}
29     else if (oprt == "/"){
30         divs = division(num1, num2);
31         cout<<"The division of "<<num1<<" and "<<num2<<" is: "<<divs;}
32     else
33         cout<<"Invalid operator";
34     return 0;
35 }
36

```

input

```

Please enter the 1st number: 10
Please enter the 2nd number: 20
Please enter the operator: +
The sum of 10 and 20 is: 30

...Program finished with exit code 0
Press ENTER to exit console.

```

```

1  #include <iostream>
2  using namespace std;
3
4  float addition(float a, float b){
5      return(a + b);}
6  float subtraction(float a, float b){
7      return(a - b);}
8  float multiplication(float a, float b){
9      return(a * b);}
10 float division(float a, float b){
11     return(a / b);}
12
13 int main()
14 {
15     float num1, num2, sum, subtr, multp, divs;
16     string oprt;
17     cout<<"Please enter the 1st number: ";cin>>num1;
18     cout<<"Please enter the 2nd number: ";cin>>num2;
19     cout<<"Please enter the operator: ";cin>>oprt;
20     if (oprt == "+"){
21         sum = addition(num1, num2);
22         cout<<"The sum of "<<num1<<" and "<<num2<<" is: "<<sum;}
23     else if (oprt == "-"){
24         subtr = subtraction(num1, num2);
25         cout<<"The difference of "<<num1<<" and "<<num2<<" is: "<<subtr;}
26     else if (oprt == "*"){
27         multp = multiplication(num1, num2);
28         cout<<"The multiplication of "<<num1<<" and "<<num2<<" is: "<<multp;}
29     else if (oprt == "/"){
30         divs = division(num1, num2);
31         cout<<"The division of "<<num1<<" and "<<num2<<" is: "<<divs;}
32     else
33         cout<<"Invalid operator";
34     return 0;
35 }
36

```

input

```

Please enter the 1st number: 10
Please enter the 2nd number: 20
Please enter the operator: \
Invalid operator

...Program finished with exit code 0
Press ENTER to exit console.

```

Report (SWE20004)

Write a one-page report on this lab covering the following:

1. Summarize the topics you explored and the activities you did during this lab.
2. Classify (group) these topics and actions under appropriate headings. Do not just copy the headings used in the instructions. For example, explain the following command, and what are the following commands do?
3. Discuss the relevance of these topics and actions in terms of C++ programming. i.e. How do the things in this lab work contribute to your understanding of C++ programming overall?
4. Why do you need to understand (and use) C++ **Functions and References**?

This report is worth 5% towards your unit assessment. Use the below page as a template. Either you can type it or write it in your words.

Introduction

This week's report will illustrate the topics covered and the activities of the lab. More specifically, we were taught about the use of functions and references. In addition, we went over some of the exercises in this week's lab in order to strengthen our knowledge and understanding of the topics.

Functions

Functions are blocks of code that are executed when they are called. Data (also known as parameters) could be passed into functions. These functions are used to carry out specific tasks and they are essential for the reuse of code. We just need to define the code once and use it multiple times.

There are several pre-defined functions, such as *main()* to execute code. You can create different functions to perform different sets of actions. Functions can be created by specifying the name of the function (whose name depends on the programmer's choice) and followed by parentheses ().

Declared functions won't execute immediately, as they are created for later usage and need to be called to be executed. Functions can be called by writing the function's name, followed by parentheses and a semicolon. There is no limitation for the number of times a function is called.

A function has two parts in it:

- **Declaration:** Contains the return type (e.g. void, int), name of the function (depends on the user) and parameters (optional).
- **Definition:** The body of the function (i.e. the code).

The declaration and the definition of a function can be separated to optimize the code.

References

A reference variable is an alternative name for an existing variable. The symbol & can be used to declare a variable as a reference. References are used for:

- Modifying the function's passed-in parameters (i.e. changing the value of a variable).
- Avoiding duplication of massive structures.
- In For Each Loop to modify all elements and avoid the copy of those elements.

Lab activities

This week's lab exercises were relatively challenging, as a large number of questions required us to fix the errors in the code in it, which could be difficult to identify the errors and even more arduous to figure out how to fix them. The errors could either be the wrong use of function or missing lines of code which are needed for the program to be executed, or both. In addition, the latter exercises require us to have some thoughts on it and although they are not as difficult as the previous questions, they are still demanding and errors could be made if we do not pay attention to the functions we are using or the variables we are declaring.

The importance of this week's topic

Knowing how to effectively declare and use multiple functions and references can help us to reduce the lines of code in the case we want to perform an action multiple times. Furthermore, knowing how to name your functions well could reduce the time and effort every time you need to call a function and understand its usage.

Conclusion

Overall, this week's lab helped us to understand the proper use of functions as well as references. I believe functions are necessary when it comes to developing software applications in C++ as many sets of actions are repeated multiple times, therefore I have to use functions in my programs more frequently to use them effectively and efficiently. I might not be very confident when using functions and references at the moment, but I opine that with more practice, I could use them easily in the future.

Name: _____ **Hong Hai Dang Nguyen** _____ **Student ID:** _____ **103503191** _____

Name: _____ **Hong Hai Dang Nguyen** _____ **Student ID:** _____ **103503191** _____