Programming Fundamental

Lab 04

Introduction

Welcome to your favorite programming Lab. In this lab manual, we shall work together to learn and implement new programming concepts.

Skills to be learned:

- Write reusable code using functions.
- Write a simple conditional statement that involves a single boolean expression
- Write a while loop to run the program continuously

Let's do some coding.

Skill: Write reusable code using functions.

Introduction

In our daily life routine, we often need to perform the same tasks again and again. For example, let's consider the example of a Car.

- Acceleration of the Car is a function that speeds up the car.
- Break in the Car is a function that stops the car.
- Turning the headlights on and off is also a function.

Functions are used to execute code repeatedly. They are the piece of the program that increases the reusability of the code. The following are the properties of functions.

- Functions are executed only when they are called
- Functions can receive different values that are known as Parameters
- Functions may or may not return a value when finished execution.

Function Prototype, function definition, and function call are the core elements of understanding a function.

The following table lists the difference between these words.

Function	Description
Function Prototype	It must be defined at the start of the program. A function prototype contains the • Return Type

Skill: Write reusable code using functions

	• Functions Name • Function Parameters
Function Definition	It is the portion of the code that implements the body/logic of the reusable functionality.
Function Call	It is the part of the code that is used to call the function inside the **main()** function. **A Function is executed only when it is called.**
Consider the following programs to get a better understanding of functions.	

Task 01(WP): Write a program that takes two numbers from the user and prints their sum on the screen.

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

int main(){

int number1, number2;
int sum;
cout<< "Enter First Number: ";
cin >> number1;
cout<< "Enter Second Number: ";
cin >> number2;
sum = number1+number2;
cout<< "Sum: " <<sum;
return 0;
}
```

Expression for calculating sum

We have written a program to solve our problem of addition.
Now, let's assume that we want to add two more numbers in this same program.
Let's modify this code to it would first take two numbers and add themes and then again takes two numbers from the user and add them too.

Skill: Write reusable code using functions

Programming Fundamental

Lab 04

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

int main(){

int number1, number2;
int sum;
cout<< "Enter First Number: ";
cin >> number1;
cout<< "Enter Second Number: ";
cin >> number2;
sum = number1+number2;
cout<< "Sum: " <<sum;

int number3, number4;
int sum1;
cout<< "Enter First Number: ";
cin >> number3;
cout<< "Enter Second Number: ";
cin >> number4;
sum1 = number4+number3;
cout<< "Sum: " <<sum1;

return 0;
}
```

Look closely and notice that **the code is being repeated**

Now, we know that functions are used for repeating structure problems.
Let's code this one out using Functions.

Task 02(WP): Write a function in a c++ program that takes two numbers from the user and prints their sum on the screen.

Now, we need to do the following to write a function.

1. **Provide Function Prototype**
2. **Provide Function Definition**
3. **Provide Function Call**

Skill: Write reusable code using functions

Programming Fundamental

Lab 04

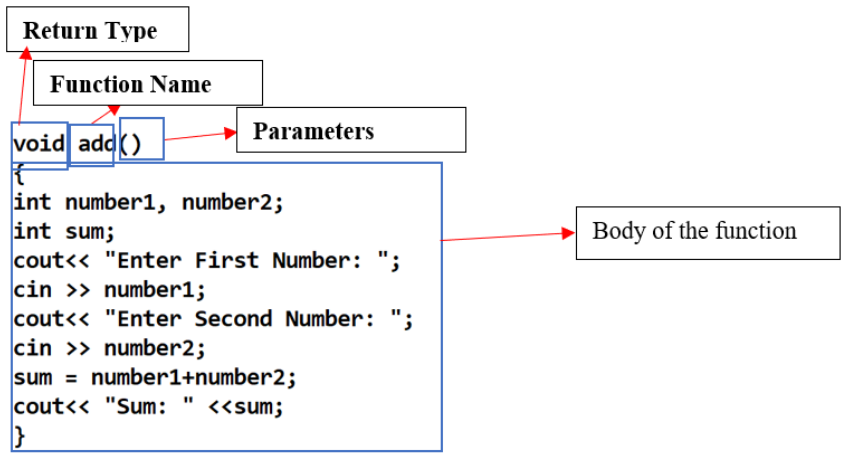
```
#include <iostream>
using namespace std;
void add();

int main(){

add();

return 0;
}

void add()
{
int number1, number2;
int sum;
cout<< "Enter First Number: ";
cin >> number1;
cout<< "Enter Second Number: ";
cin >> number2;
sum = number1+number2;
cout<< "Sum: " <<sum;
}
```

This is **Function Prototype** that consists of **return type, function name, and parameters**

This is the **function Call**

This is the **function Definition**. This is where **we define the functionality of the function** after writing the function prototype

See the decomposition of a function into different parts that constitute a function.

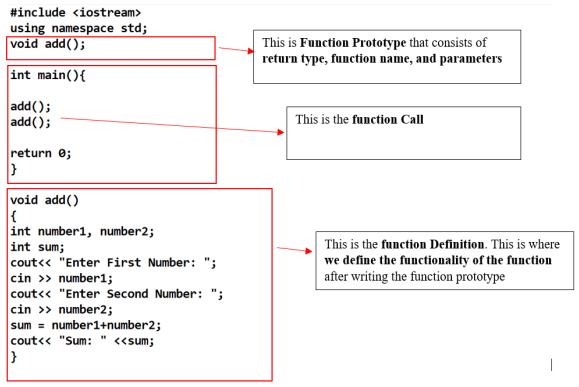


Now, we can just **call the function as many times as we want.**
Look at the below code and output as an example:

Skill: Write reusable code using functions

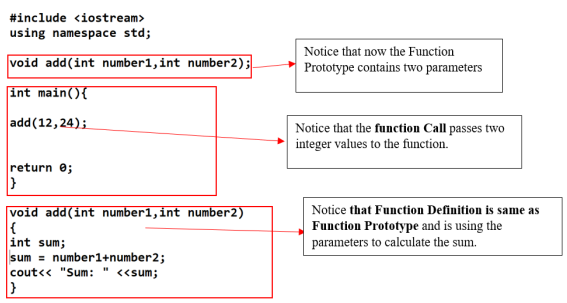
Programming Fundamental

Lab 04




```
D:\PF codes>c++ test1.cpp -o test.exe

D:\PF codes>test.exe
Enter First Number: 34
Enter Second Number: 12
Sum: 46Enter First Number: 56
Enter Second Number: 48
Sum: 104
D:\PF codes>
```

Great Work Guys !! You have implemented your first program using Functions in c++.
Let's learn about functions with parameters now.
Task 03(WP): Write a function in a c++ program that **takes two numbers as parameters** and prints their sum on screen.




```
D:\PF codes>c++ test1.cpp -o test.exe

D:\PF codes>test.exe
Sum: 36
D:\PF codes>
```

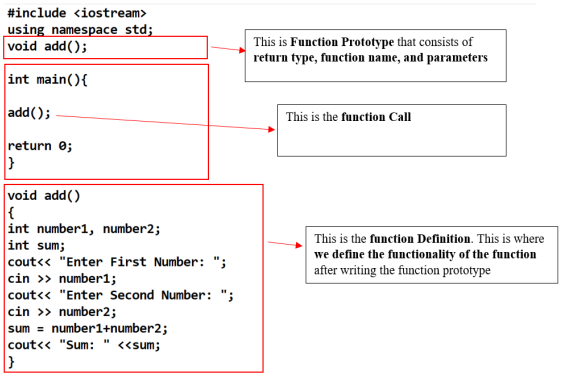
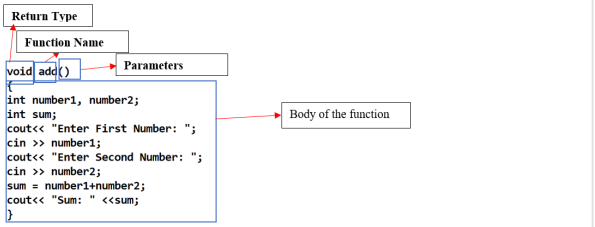
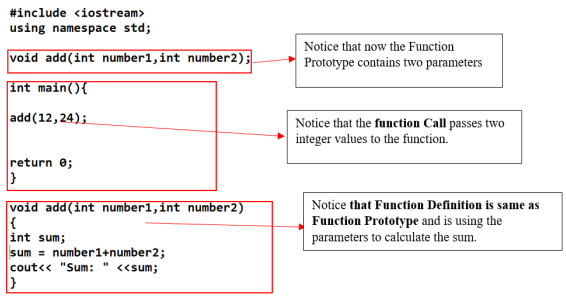
Now, the above-mentioned screenshots define a function that **takes two integer values as parameters and prints their sum on the screen.**
Notice that most of the code is the same, we have just provided the function two values at the time of the function call instead of taking two values from the user.

Skill: Write reusable code using functions

Programming Fundamental

Lab 04

Conclusion:

Functions are the pieces of code that are used to solve repeating structure problems. To implement functions, you need a **function prototype, function definition, and function call**	
Function prototype is repeated in the function definition and consists of three elements: return type, function name, and parameters	
Functions can take values with them to perform different operations on them. These are called parameters.	
Well done Students, You have just learned another skill.	

Task 01(OP): A person is eligible to vote if his/her age is greater than or equal to 18. Define a function that prints if he/she is eligible to vote.

Skill: Write reusable code using functions

Programming Fundamental

Lab 04

Skill: Write a simple conditional statement that involves a single boolean expression

Introduction

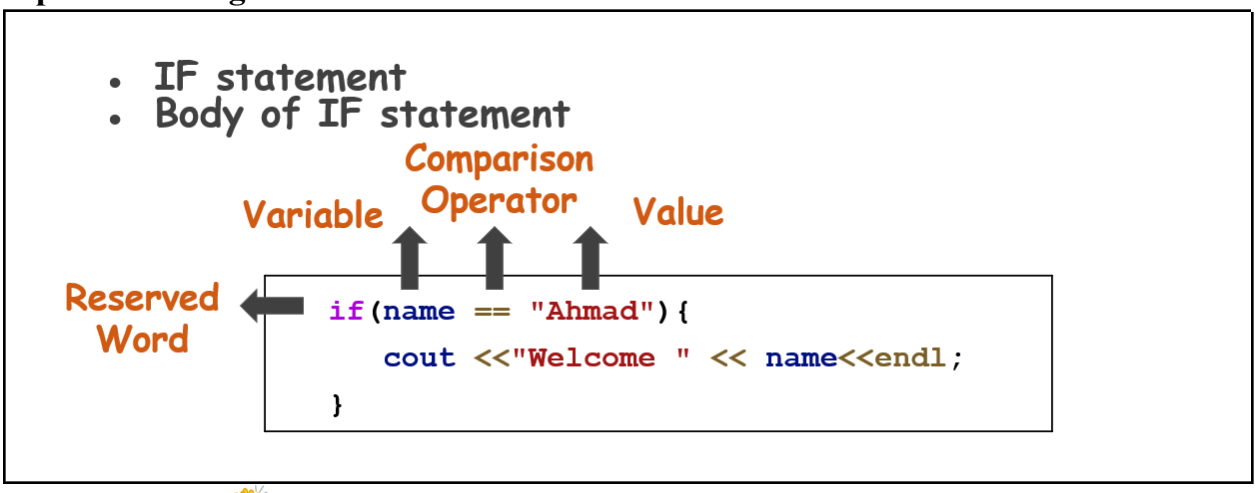
Conditional Statements are used to execute the code depending on some given conditions. In our daily life, we encounter many situations where we have to decide from multiple options. For example, whether to buy a dress or not? Or Whether to go to university today or not?

Even simple cases present us with these potential solutions that can be chosen depending on different conditions. For example, IF the cost of the dress is more than 1500 Then you will buy that dress. OR IF your friends are going to university Then you will go to the university😀.

Similarly, we are faced with similar problems in programming as well. Where we, as programmers, have to write programming code to solve such problems. Therefore, for such problems, we make use of the IF Statement.

IF Statement

We learned about solving such issues in the last class through various examples. For example, consider the below-mentioned cell where we are **evaluating a boolean expression using an IF Statement.**



Most Important 👋

A Boolean Statement evaluates to either **True or False**. If it's **true** the body of the IF Block is **executed** however, if the boolean expression evaluates to **False**, the code written inside the curly braces is ignored and **not executed** by the compiler.

Skill: Write a simple conditional statement that involves a single boolean expression

Programming Fundamental

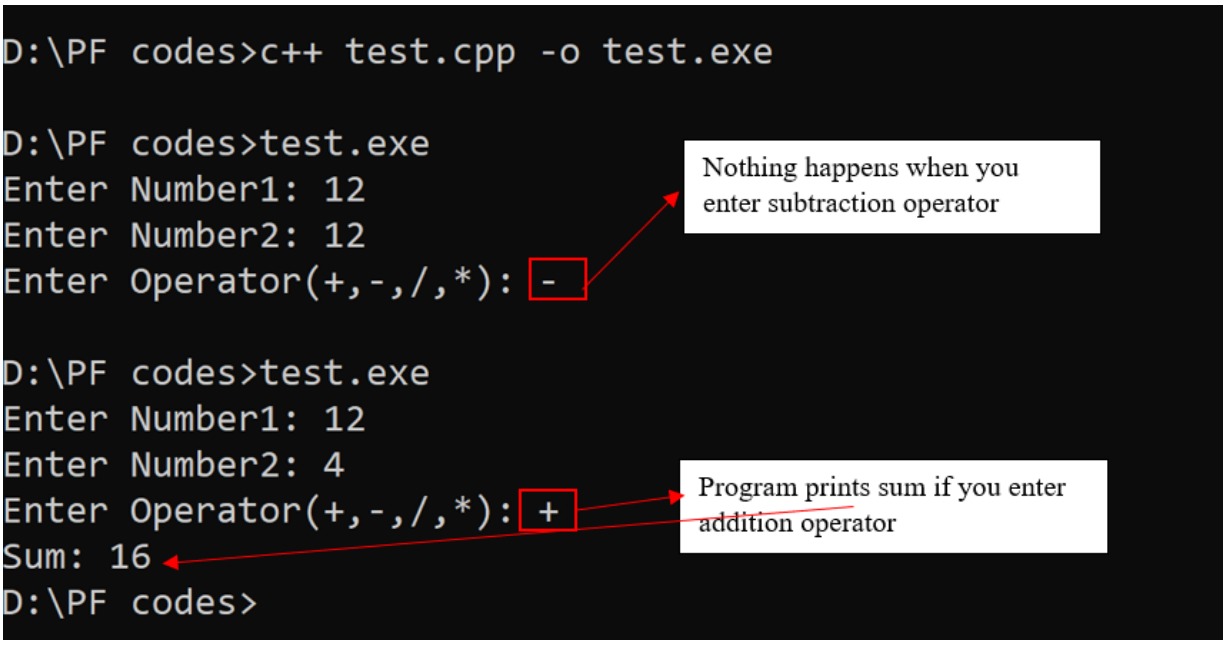
Lab 04

Let's code it out !

Consider the following task.

Task 01(WP): Write a function that takes a **mathematical operator from the user** and then **asks to enter two numbers**. If the operator is an **addition operator (+)** the function should **add the numbers and print their sum on the screen** otherwise nothing happens.

Sample output:



Now, take a minute and **think about how it can be done by using a boolean expression inside an IF Block**.

Skill: Write a simple conditional statement that involves a single boolean expression

Programming Fundamental

Lab 04

```
#include <iostream>
using namespace std;
void add(int number1, int number2);
int main(){
int number1, number2;
char op;
cout<< "Enter Number1: ";
cin >> number1;
cout<< "Enter Number2: ";
cin >> number2;
cout<< "Enter Operator(+,-,/,*): ";
cin >> op;
if(op == '+'){
        add(number1,number2);
}
return 0;
}
void add(int number1, int number2)
{
cout<< "Sum: " << number1+number2;
}
```

1 → Define a function prototype that will take two numbers as input

2 → Declare two integer and a character type variable and take input from the user and store

3 → Write an IF Block that if **True** executes the function call

4 → Provide the function Definition and Ask the user to enter two numbers.

You are doing great so far. Great Work People!

Task 02(OP): **Modify** the above program that prints the **respective result depending on what mathematical operator the user has entered**. Implement for **division, multiplication, subtraction, and addition**.
Hint: You will need to write multiple IF Blocks.

Conclusion

Statement	Description
IF Block	Reserve Word: `if` — `(op == '+')` → Boolean expression that evaluates to either True or False. `{ cout<< number1+number2; }` → It is the Body of the IF Block executed only if the Boolean expression is True. Otherwise, it is not executed by compiler

Skill: Write a simple conditional statement that involves a single boolean expression

Programming Fundamental

Lab 04

Task 03(OP): Write a program that inputs a number from the user and outputs pass if the user has entered a number greater than 50 and fails otherwise.

Task 03A(OP):

Instruction: Define a function that accepts a parameter (marks) from the user and print the result on the screen.

Task 04(OP): Write a program that inputs a number from the user and prints **Even** if it's an even number and prints **Odd** if it's an odd number.

Task 04A(OP):

Instruction: Define a function that takes a parameter (number) from the user and prints the result on the screen.

Task 05(OP): Write a program that inputs two numbers from the user and prints the **greater** number.

Task 05A(OP):

Instruction: Define a function that takes two parameters from the user and prints the larger number on the screen.

Task 06(CP): A Store has announced to give a 10% **discount** on the total purchase amount every Sunday. Write a program that takes Day, and total purchase amount, and outputs the payable amount.

Task 06A(CP):

Define a function that takes two parameters (as Day and Total Purchase amount) from the user and prints the payable amount.

Skill: Write a simple conditional statement that involves a single boolean expression

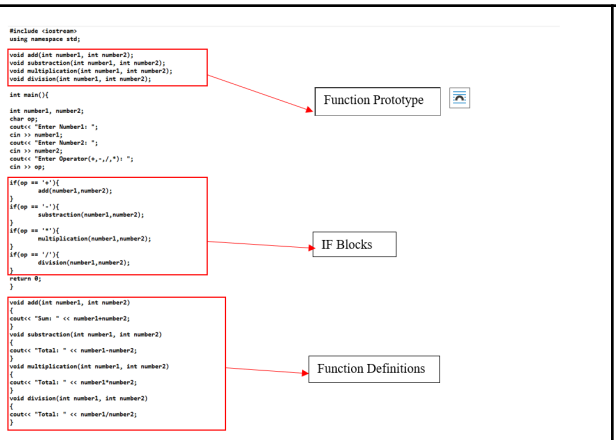
Programming Fundamental

Lab 04

Skill: Write a while loop to run the program continuously

Introduction

Let's just look at the progress of our program that we have been able to implement from the start of this Lab.




```
D:\PF codes>c++ test.cpp -o test.exe

D:\PF codes>test.exe
Enter Number1: 12
Enter Number2: 32
Enter Operator(+,-,/,*): +
Sum: 44
D:\PF codes>test.exe
Enter Number1: 2
Enter Number2: 3
Enter Operator(+,-,/,*): *
Total: 6
D:\PF codes>test.exe
Enter Number1: 32
Enter Number2: 12
Enter Operator(+,-,/,*): -
Total: 20
D:\PF codes>test.exe
Enter Number1: 6
Enter Number2: 2
Enter Operator(+,-,/,*): /
Total: 3
D:\PF codes>
```

We have a working simple calculator but the problem is that after each step **we need to execute the program again** in order to perform another operation.

If only the program could execute itself again on its own.

Thinking about Solution ?

Loops are used to execute the program repeatedly under a given condition.

Any code written **inside these curly braces will be executed an infinite number of times**.	`while(true)`
	`{`
	
	`}`

Let's add this functionality into the above mentioned code to make it repeat itself.

Question !

Which code do you think that we need to put inside these curly braces so it will be executed again.

Think again ! You may be surprised.

Hint: After printing the result on screen we want the program to again ask for input and then execute the function call depending on the entered operator.

Skill: Write a while loop to run the program continuously

Programming Fundamental

Lab 04

Task 01(WP): Write a program using functions and if statements that **repeatedly** acts like a **calculator** and **never stops** unless we **forcefully** close the window.

```
void add(int number1, int number2);
void substraction(int number1, int number2);
void multiplication(int number1, int number2);
void division(int number1, int number2);

int main(){

int number1, number2;
char op;
while(true)
{
cout<< "Enter Number1: ";
cin >> number1;
cout<< "Enter Number2: ";
cin >> number2;
cout<< "Enter Operator(+,-,/,*): ";
cin >> op;

if(op == '+'){
        add(number1,number2);
}
if(op == '-'){
        substraction(number1,number2);
}
if(op == '*'){
        multiplication(number1,number2);
}
if(op == '/'){
        division(number1,number2);
}
}
return 0;
}

void add(int number1, int number2)
{
cout<< "Sum: " << number1+number2<<endl;
}
void substraction(int number1, int number2)
{
cout<< "Total: " << number1-number2<<endl;
}
void multiplication(int number1, int number2)
{
cout<< "Total: " << number1*number2<<endl;
}
void division(int number1, int number2)
{
cout<< "Total: " << number1/number2<<endl;
}
```

All the code written inside the curly braces of while(true) will be executed indefinite times.

```
D:\PF codes>c++ test.cpp -o test.exe

D:\PF codes>test.exe
Enter Number1: 12
Enter Number2: 3
Enter Operator(+,-,/,*): *
Total: 36
Enter Number1: 12
Enter Number2: 12
Enter Operator(+,-,/,*): +
Sum: 24
Enter Number1:
```

Skill: Write a while loop to run the program continuously

Programming Fundamental

Lab 04

Task 02(OP): Write a program that keeps **printing your name** until closed **forcefully**.
Task 03(OP): Write a program that keeps printing the **name entered by the user**.
Instruction: Use a **function** that accepts a **string parameter** name and prints that name
Task 04(OP): Write a program that keeps printing the name entered by the user **only if the name is "Irzam" otherwise the program should again ask the user to enter the name**.
Instruction: Use function that accepts a string parameter name and prints that name
Task 05(CP): A Store has announced to give a 10% discount on the total purchase amount of every Sunday and 5% on every other day. Write a program that takes Day, total purchase amount and outputs the payable. Now, After outputting the payable amount on screen, the program should again ask for the same details of some other customer and go on until closed forcefully.

You have learnt a lot today. Keep practicing these codes and we will see you on programming day with new skills to learn.

Good Luck and Best Wishes !!

Happy Coding ahead :)

Skill: Write a while loop to run the program continuously