



# Programming Fundamental

## Lab Manual 2



---

### Learning Outcomes:

- Students should be able to think logically and develop problem-solving skills
- Students should be able to learn how to declare variables and manipulate their values
- Students should be able to write the input and output instructions

### Introduction

In this class, we shall learn about “program structure”, “variables”, “input”, and “output” concepts in c++ that are used to solve different mathematical problems.

### Program structure

Just like any other language, the c++ language also has a particular structure that requires that instructions must be given according to the defined structure.

A standard program structure for the c++ programming language is given below.

```
#include<iostream>
```

```
using namespace std;
```

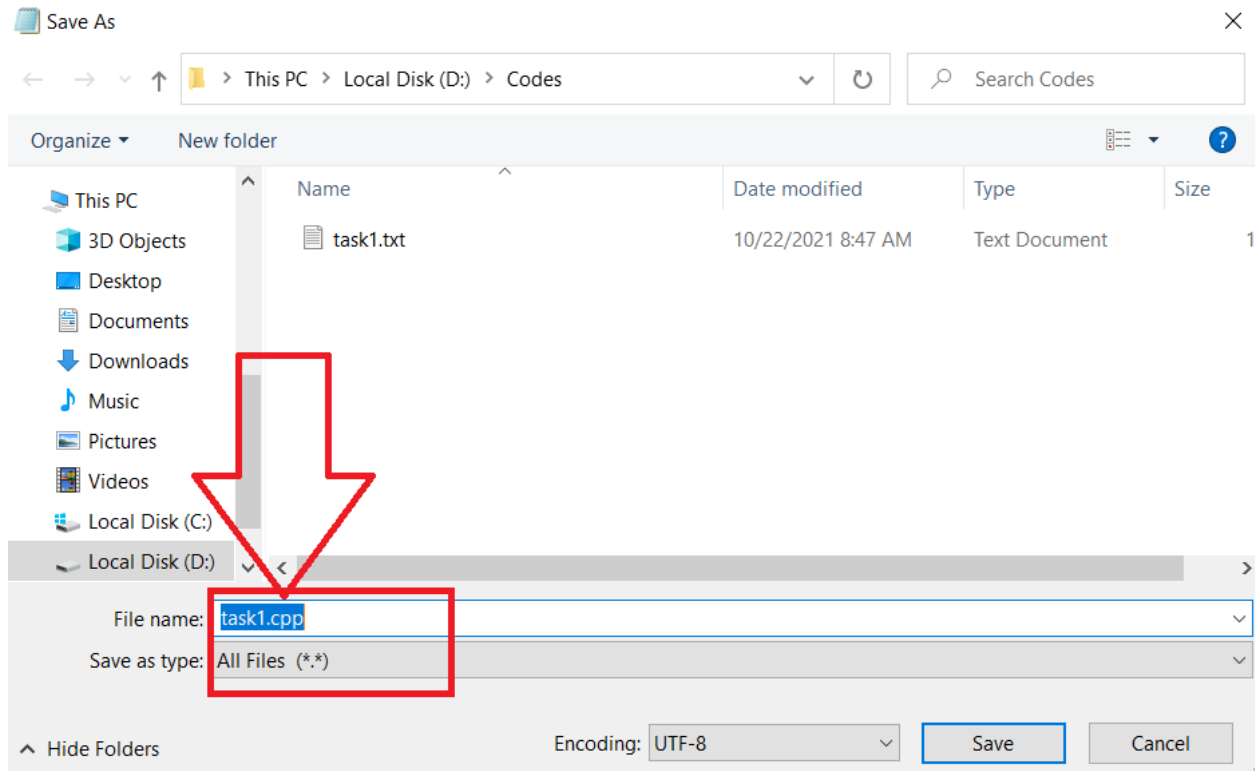
```
int main()
```

```
{
```

```
}
```



The c++ language rules require that all the instructions must be given inside the highlighted section and the file save the file through the “**Save as**” option with “**.cpp**” at the end of the file name.



**Activity:** Write these lines of code in your editor (notepad) to graph the practical understanding of the basic program structure of the c++ language.

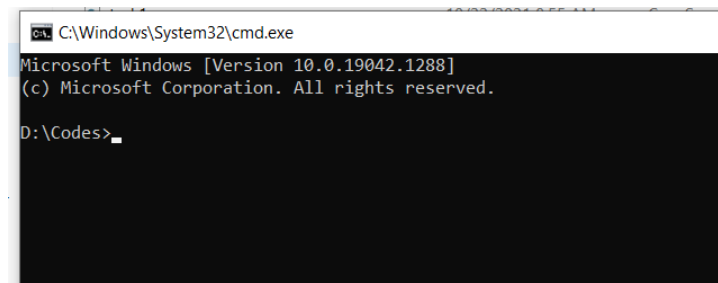
## Compiler

Initially, we shall use the windows command-line interface to compile and execute our programs. To execute the program, perform the following steps.

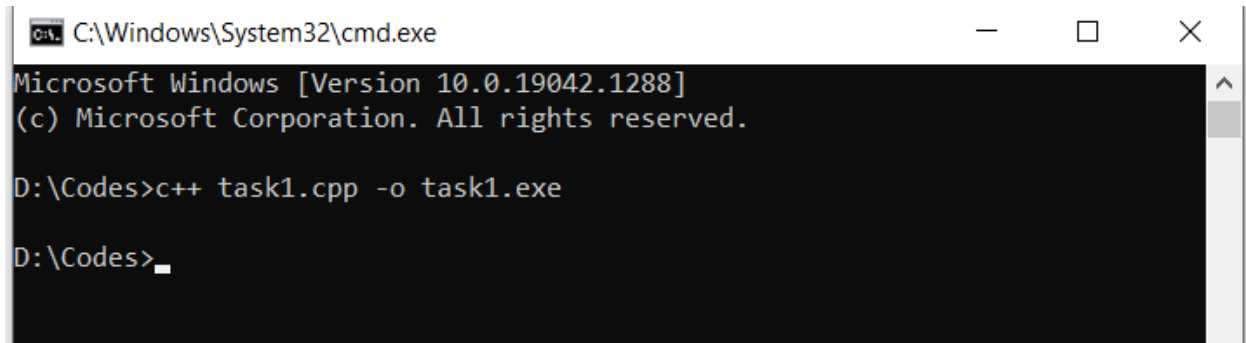
Open the command prompt in the directory in which you have stored the C++ code file.



Click on the address bar and write “cmd” and press enter.



Write the following line in the cmd prompt in exact order but replace “**task1.cpp**” with your code file and “**task1.exe**” with the name of the file that you want to create for your program.

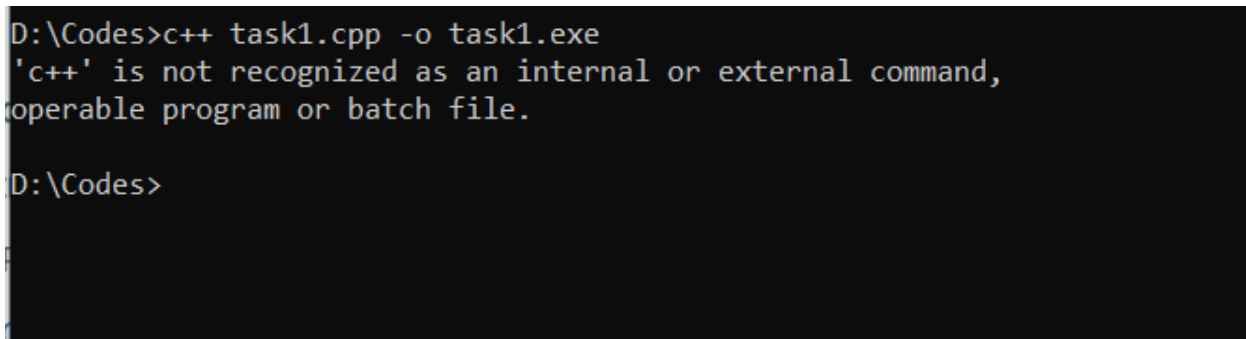


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

D:\Codes>c++ task1.cpp -o task1.exe

D:\Codes>_
```

[If you get the error, then go-to activity on the last to install the compiler mingw and set the environment path variables]

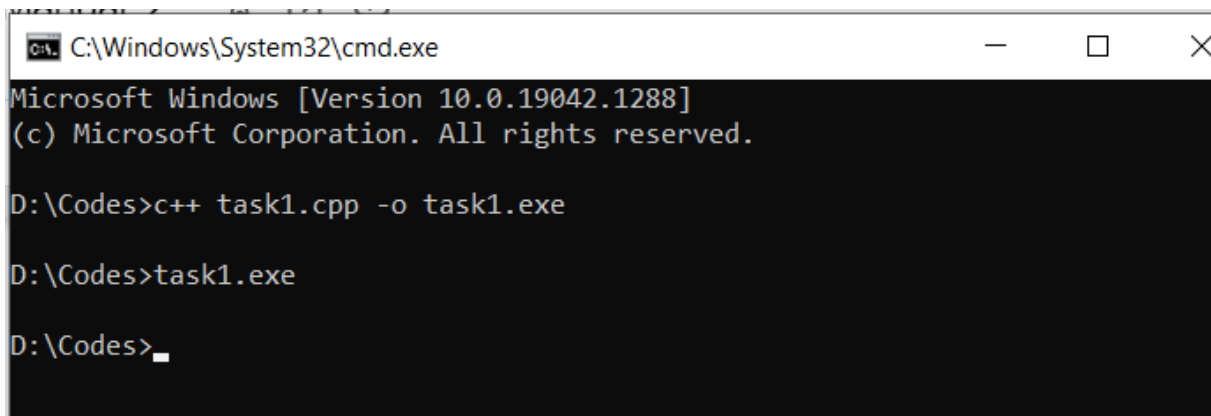


```
D:\Codes>c++ task1.cpp -o task1.exe
'c++' is not recognized as an internal or external command,
operable program or batch file.

D:\Codes>
```

This is the error you would see on the screen. If your program is compiled successfully, move onwards.

Now, you can execute this code by typing the name of the file that you have written at the end of the previous instruction.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

D:\Codes>c++ task1.cpp -o task1.exe

D:\Codes>task1.exe

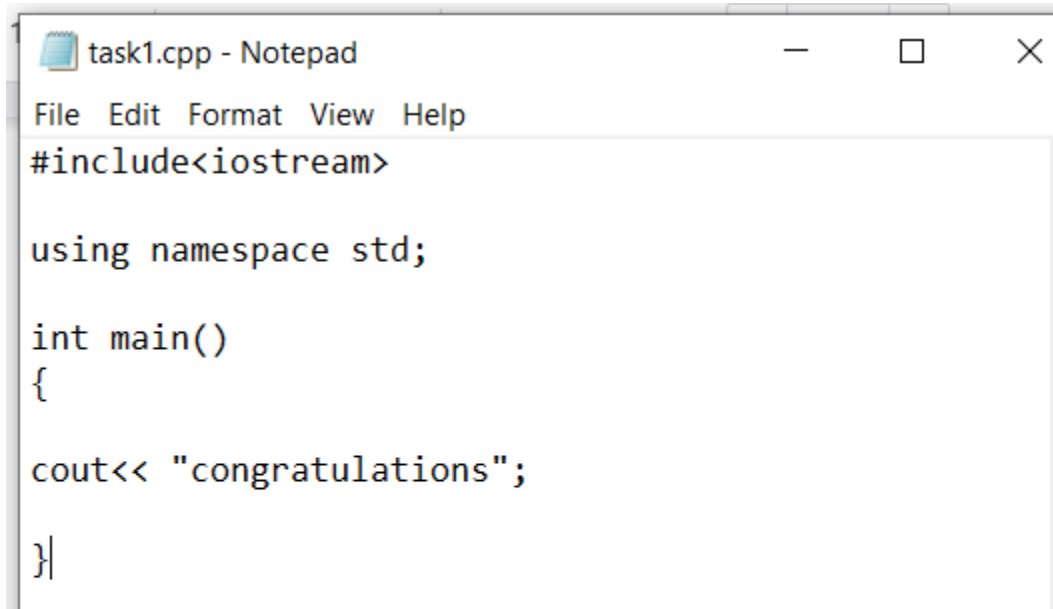
D:\Codes>_
```

Observe that the file has been successfully executed and output has been displayed.

## Output Instruction

Now add the following instruction in your program.

```
cout<< "Congratulations";
```



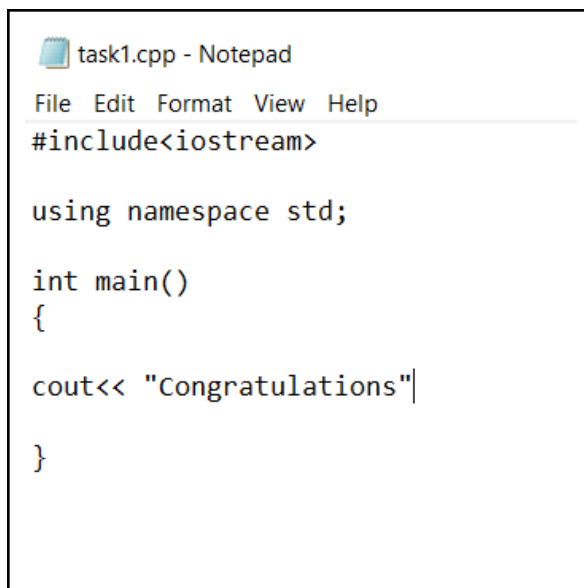
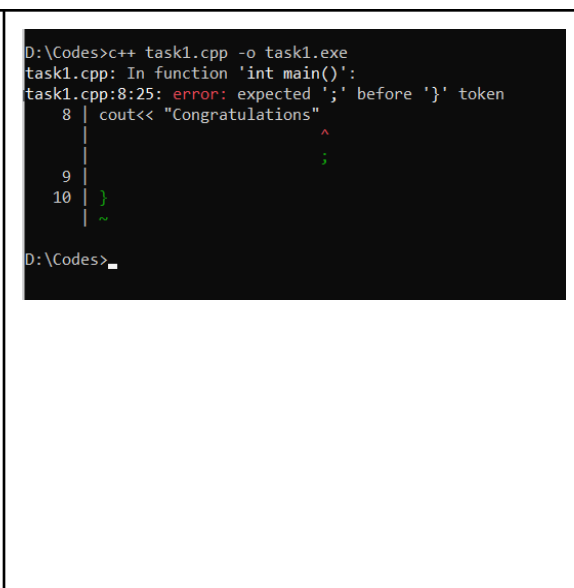
```
task1.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{
    cout<< "congratulations";
}
```

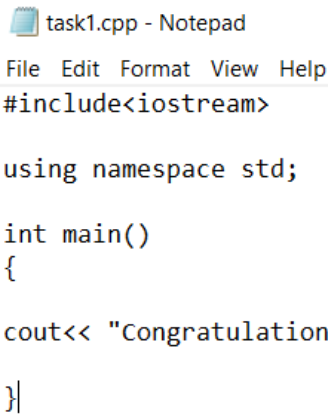
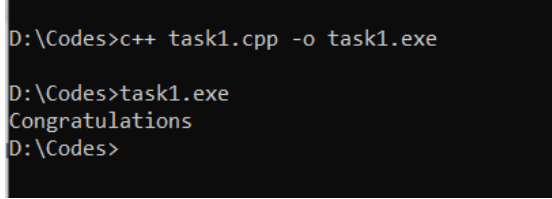
Finally, as in English, each sentence is ended with a **full stop** (.) at the end, similarly, in C++ we use a **semicolon** (;) at the end to tell that the sentence (instruction) has ended.

Let's observe the difference of execution of a program line without “;” at the end of the instruction.

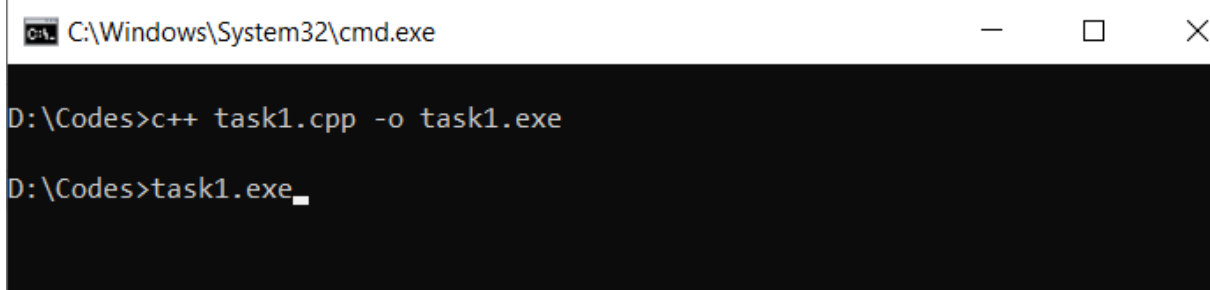
 <pre>task1.cpp - Notepad File Edit Format View Help #include&lt;iostream&gt;  using namespace std;  int main() {     cout&lt;&lt; "Congratulations" }</pre>	 <pre>D:\Codes&gt;c++ task1.cpp -o task1.exe task1.cpp: In function 'int main()': task1.cpp:8:25: error: expected ';' before '}' token    8   cout&lt;&lt; "Congratulations"                             ^    9     10   }        ~ D:\Codes&gt;_</pre>
---	---

We have not added “;” at the end of <b>Cout</b> statement	Observe that the program has not been successfully compiled and generated an <b>error</b> .
---	---

Now, Let's add the “;” at the end of instruction as well to observe the difference in execution.

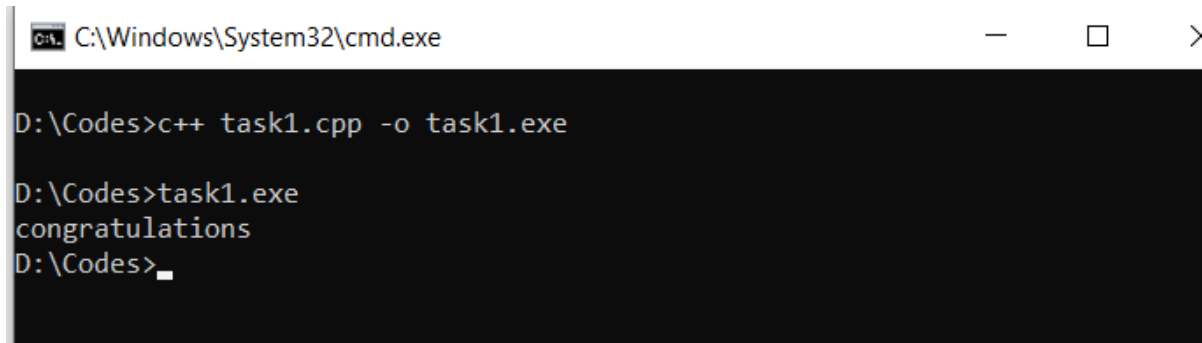
 <pre>task1.cpp - Notepad File Edit Format View Help #include&lt;iostream&gt;  using namespace std;  int main() { cout&lt;&lt; "Congratulations"; }</pre>	 <pre>D:\Codes&gt;c++ task1.cpp -o task1.exe D:\Codes&gt;task1.exe Congratulations D:\Codes&gt;</pre>
We have added the “;” at the end of our first variable instruction.	Now, observe that the program has been <b>compiled</b> successfully and also completed its execution.

Now open the cmd and follow the previous procedure once again to **compile and execute** the program.



```
C:\Windows\System32\cmd.exe
D:\Codes>c++ task1.cpp -o task1.exe
D:\Codes>task1.exe_
```

Observe that the program has been successfully executed and produced the following output.



```
C:\Windows\System32\cmd.exe

D:\Codes>c++ task1.cpp -o task1.exe

D:\Codes>task1.exe
congratulations
D:\Codes>_
```

**“Congratulations, You have successfully created, compiled, and executed your first program”**

## Variable and Types

In programming, there are “nicknames” that are known as “**variables**” that are used to store different kinds of values.

For example,

Ali had obtained 850 marks in matric. We want to store this number in computer memory. But, the problem is we can not possibly remember the address of all the things that we stored in the memory. That’s why we use nicknames, “variables”.

So we can use a variable to store the obtained marks of Ali in the storage of the computer.

For example, we can name the variable as “**marks**”.

### integer datatype:

However, as obtained marks would be an **integer number**, we shall use the integer datatype for storing this value into the variable.

In c++, the integer data type is denoted by “**int**”.

Let’s add the data type to our first variable.

**int marks**

Similarly, Let’s add the “;”(instruction terminator) to our first c++ variable declaration as well.

**int marks;**

**Activity: Write your first line of c++ language on your computer editor by declaring a variable.**



```
test.txt - Notepad
File Edit Format View Help
#include<iostream>

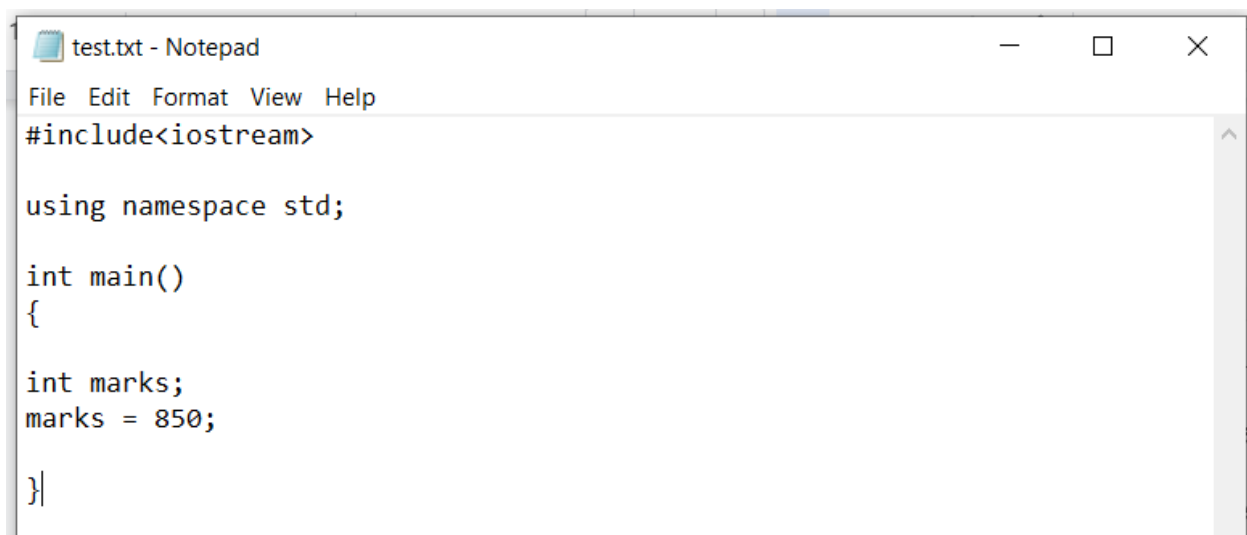
using namespace std;

int main()
{
    int marks;
}
```

**“Congratulations, you have successfully learned how to declare a variable”.**

Now, we can assign any natural number to this marks variable. In this case, let's assume that Ali has obtained 850 marks.

**marks = 850;**



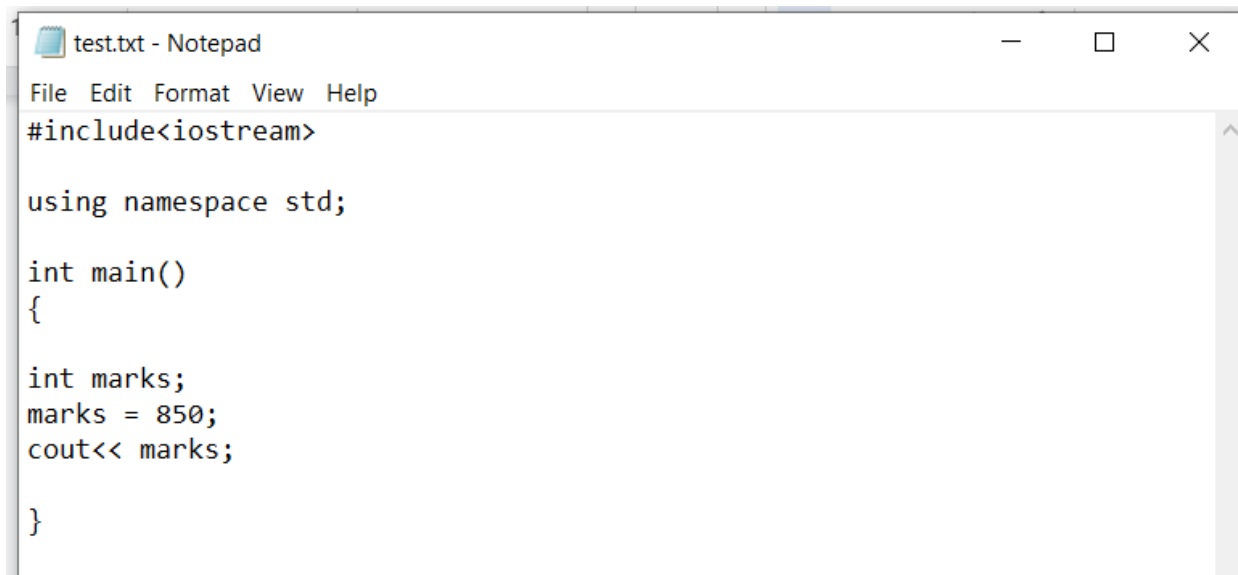
```
test.txt - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{
    int marks;
    marks = 850;
}
```

Now let's check the value that is stored in this variable using the below instruction.

**cout<< marks;**



```
test.txt - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

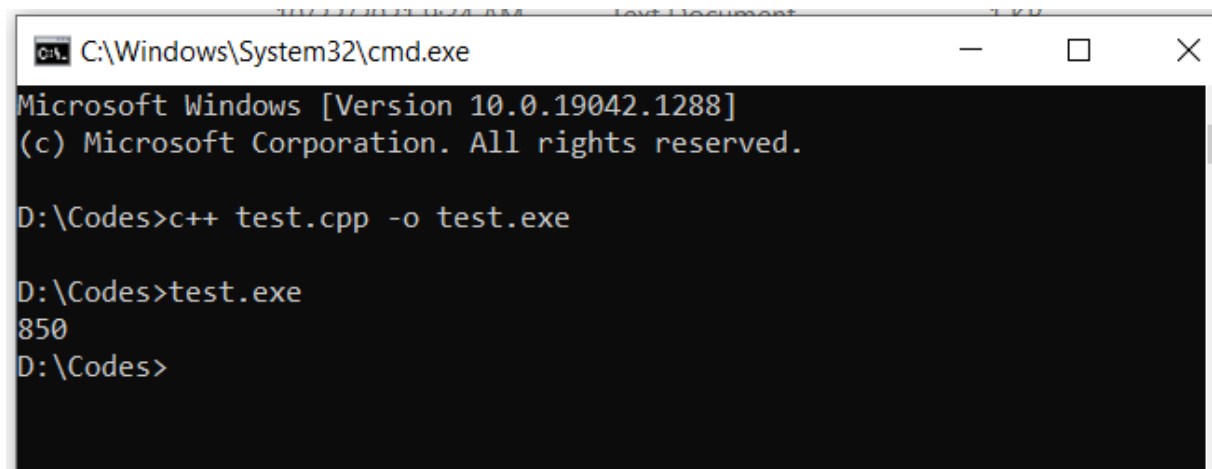
int main()
{
    int marks;
    marks = 850;
    cout<< marks;
}
```

This code will tell us the value stored in the “**marks**” variable.

Note: This instruction of “**cout<< marks**” shall be explained in the coming section of the document.

Now “**compile and execute**” the program using the cmd to observe the changes in the program.

The program should produce the following output.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
850
D:\Codes>
```

**Activity:** Write The above code on your computer editor to check the value stored in the variable.

### Example:

Similarly, we can also declare a variable for storing total marks in this way.

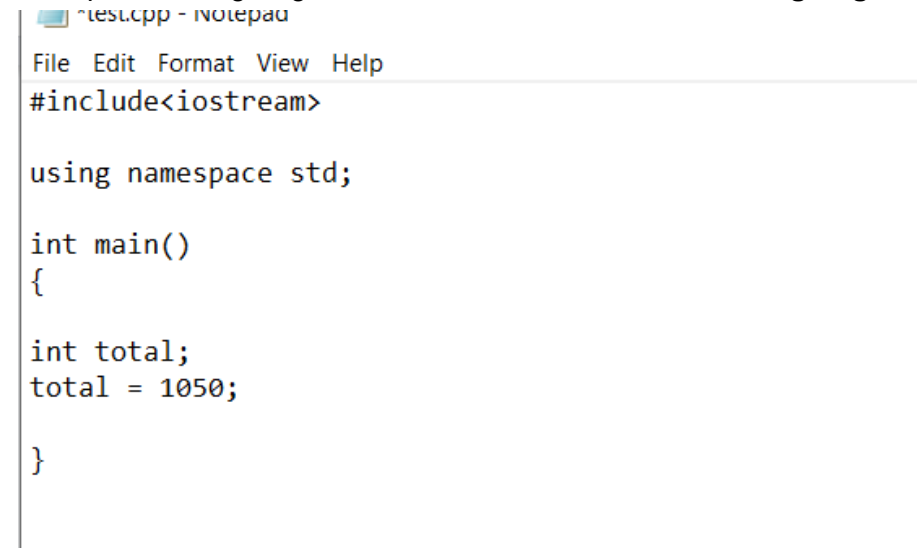
**int total;**

Now, similarly, we can store total numbers in this variable.



**total = 1050;**

“This process of giving value to a variable is known as “**assigning value to variable**”.



```
~test.cpp - Notepad++
File Edit Format View Help
#include<iostream>

using namespace std;

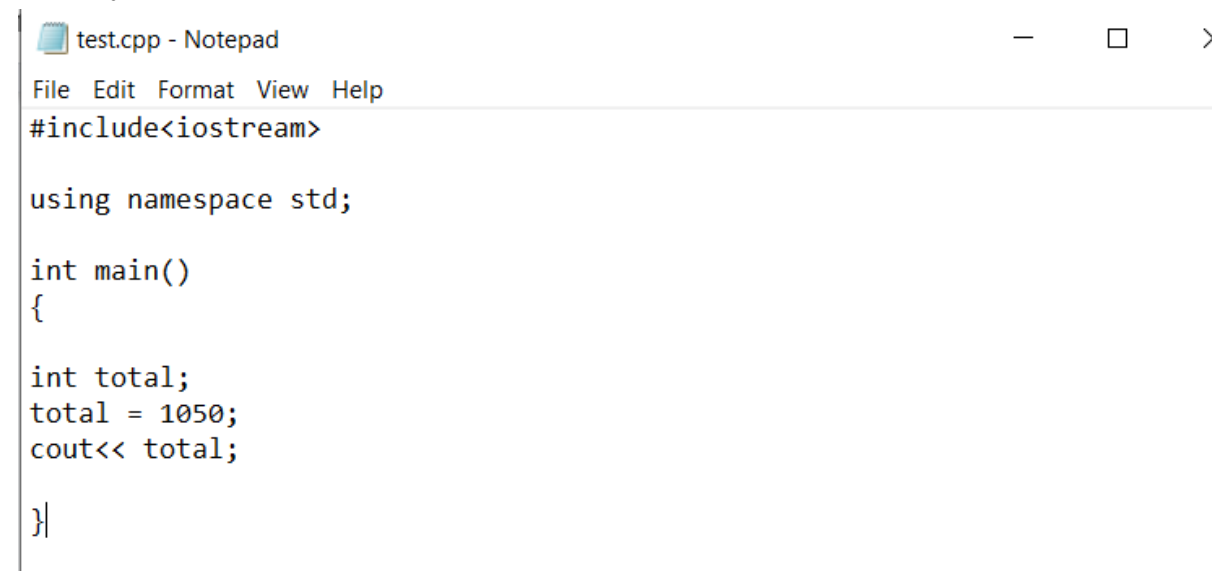
int main()
{

int total;
total = 1050;

}
```

**Activity: Write the above code on your computer editor to store the value in the variable.**

Similarly, let's also check the value stored in the total variable.



```
test.cpp - Notepad
File Edit Format View Help
#include<iostream>

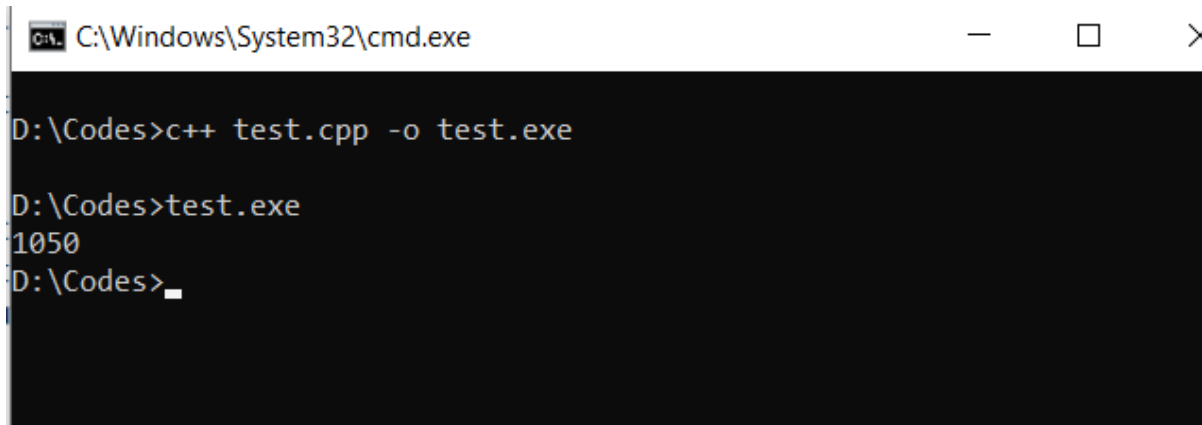
using namespace std;

int main()
{

int total;
total = 1050;
cout<< total;

}
```

**The output of this code is as follows.**



```
C:\Windows\System32\cmd.exe

D:\Codes>c++ test.cpp -o test.exe

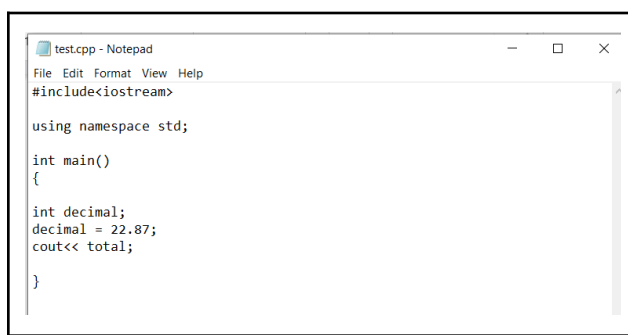
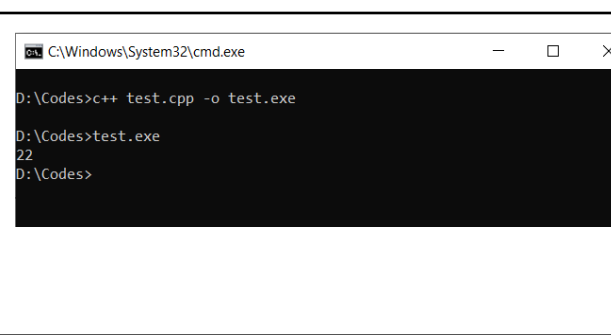
D:\Codes>test.exe
1050
D:\Codes>
```

**Activity:** Write The above code on your computer editor to check the value stored in the variable.

### Float Datatype:

What if we wanted to store the percentage of the marks in the storage of the computer. These decimal values can not be stored in “int” datatype as “int” datatype can only store natural numbers.

To develop a better understanding of this concept, let's first try to store “**a decimal value**” into an “**int**” type variable.

 <pre>test.cpp - Notepad File Edit Format View Help #include&lt;iostream&gt;  using namespace std;  int main() {     int decimal;     decimal = 22.87;     cout&lt;&lt; total; }</pre>	 <pre>C:\Windows\System32\cmd.exe  D:\Codes&gt;c++ test.cpp -o test.exe  D:\Codes&gt;test.exe 22 D:\Codes&gt;</pre>
<b>Activity:</b> Write this code in your computer editor and observe the difference in produced output.	<b>Activity:</b> Compile and execute the program.

You might have observed the value “22. 87” is not printed on the computer screen. Meaning the “**int**” datatype can not store the decimal point value. And therefore, for this purpose we use a different data type named as “**float**” datatype.

Let's use this data type to store a decimal value into our program, for example, the percentage of a student's marks.

```
float percentage;  
percentage = 70.78;
```

```
#include<iostream>  
  
using namespace std;  
  
int main()  
{  
  
float percentage;  
percentage = 70.78;  
  
}
```

**Activity:** Write the above code on your computer editor (notepad) to store the value in the float type variable.

**Example:**

declare a variable named as “**number**” to store a value of 400.6.

```
float number;  
number = 400.6;
```

```
#include<iostream>  
  
using namespace std;  
  
int main()  
{  
  
float number;  
number = 400.6;  
  
}
```

**Activity:** Write the above code on your computer editor to store the value in the variable.

You can also check the value stored in float type variables by using the “**cout<<** “ instruction.

```
#include<iostream>

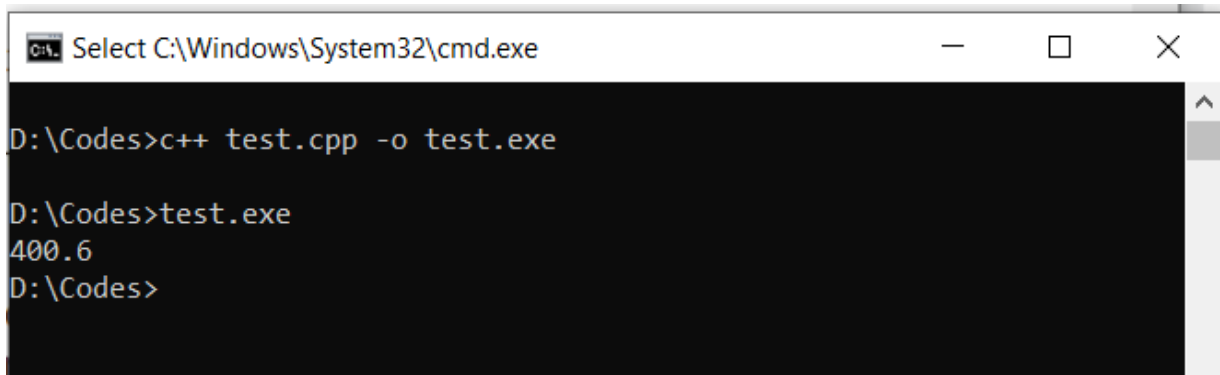
using namespace std;

int main()
{

float number;
number = 400.6;
cout<< number;

}
```

The output of this code is as follows.



```
C:\> Select C:\Windows\System32\cmd.exe

D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
400.6
D:\Codes>
```

**Activity:** Write The above code on your computer editor to check the value stored in the variable.

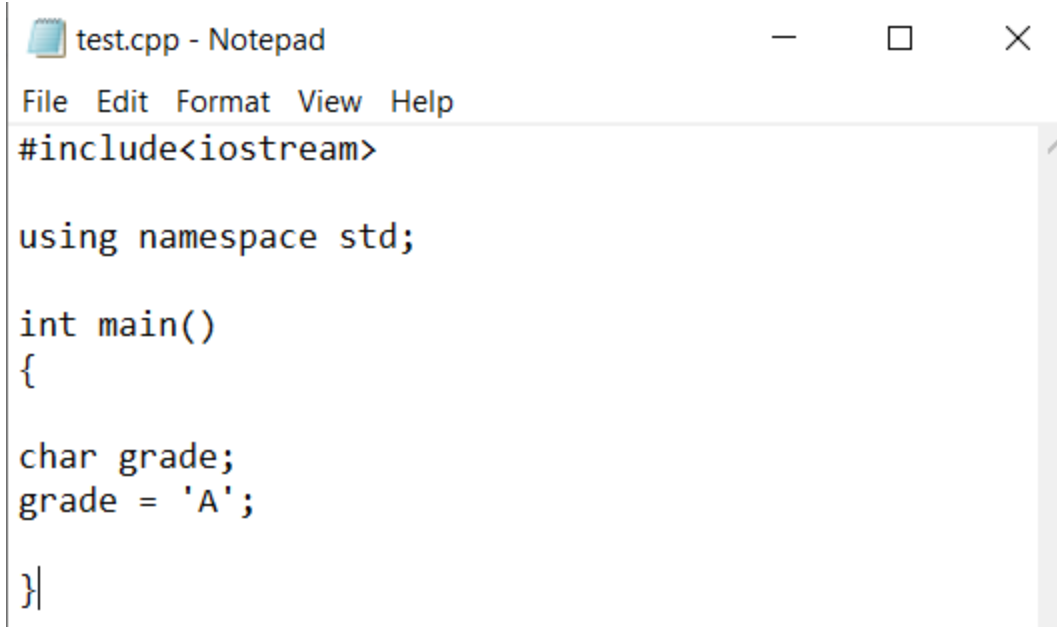
### Char Datatype:

What if we wanted to store the grade of the marks as well in the storage of the computer. These single character values can not be stored in “int” or “float” data type as “int” and “float” datatypes can only store natural and decimal numbers respectively.

For this purpose, the c++ language has a data type called “**char**”.

Let’s use this data type to store the grade of the student’s marks.

```
char grade;
grade = 'A';
```



```
test.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{

char grade;
grade = 'A';

}
```

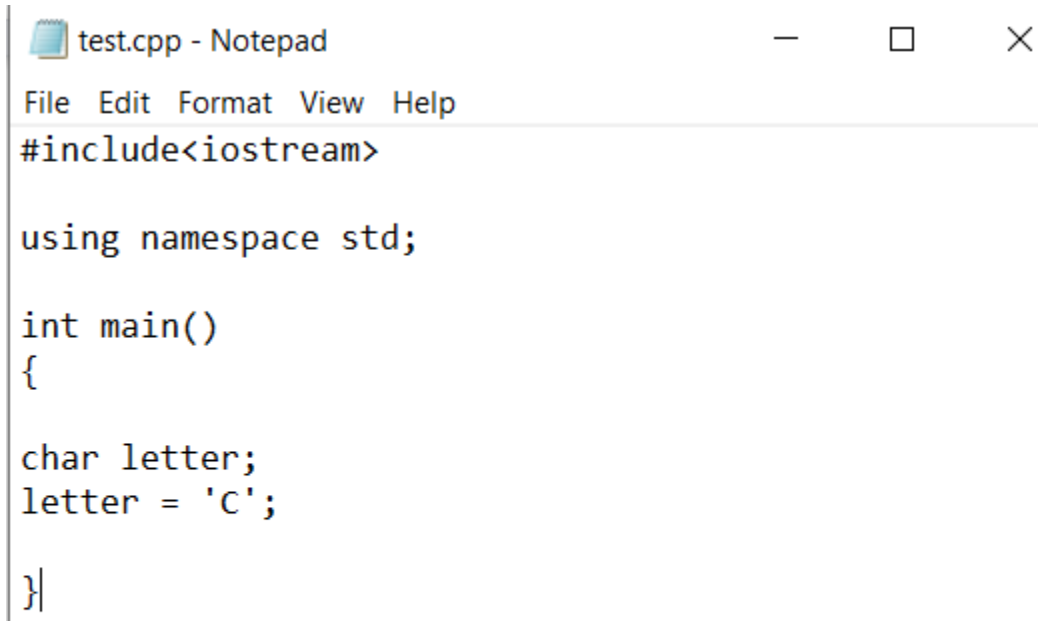
**Activity:** Write the above code in your computer to declare a char type variable in your program.

**Example#2:**

Declare a variable named “letter” to store “C”.

**char letter;**

**letter = 'C';**



```
test.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

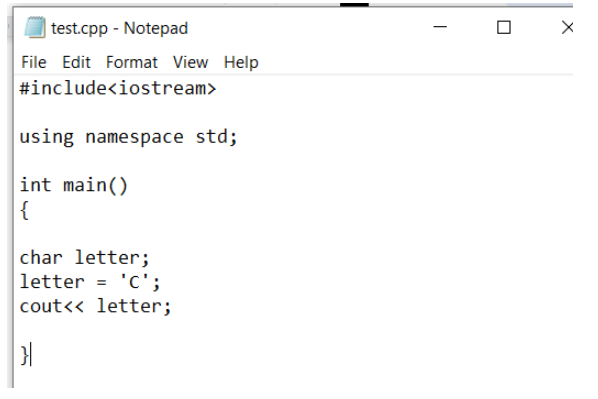
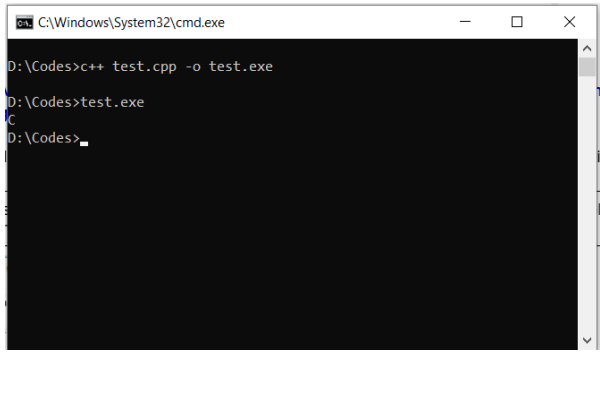
int main()
{

char letter;
letter = 'C';

}
```

**Activity:** Write the above code in your computer to declare a char type variable in your program.

Similarly, you can also get the values of these variables by using the “**cout<<**” instruction.

Lets add the cout instruction into our computer.	The output of the above code is as follows
	

**Activity:** Write The above code on your computer editor to check the value stored in the character variable.

### String datatype:

What if we wanted to store the name of the student as well in the storage of the computer. These string values can not be stored in “char” datatype as “char” datatype can only store single character values.

For this purpose, the c++ language has a data type called “**string**”.

Let’s use this data type to store the name of the student.

```
string name;  
name = “Ali”;
```

```
#include<iostream>

using namespace std;

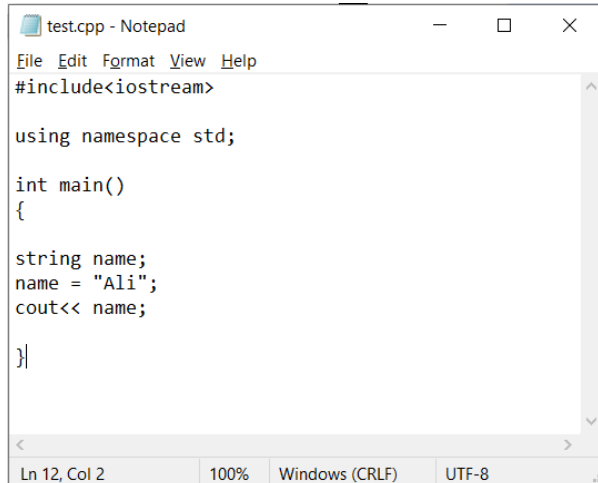
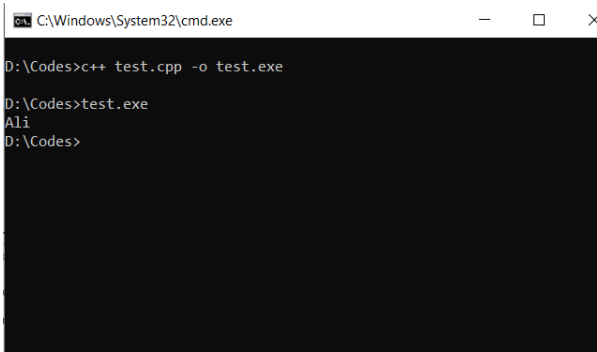
int main()
{

string name;
name = "Ali";

}
```

**Activity:** Write the above code in your computer editor to declare a string-type variable in your program.

You can check the value stored in this variable by using the “**cout<<**” instruction.

<p>Lets add the <b>cout</b> instruction into our program.</p>	<p>Output of the program is as follows.</p>
 <p>The screenshot shows a Notepad window titled 'test.cpp - Notepad'. The code is the same as above but with 'cout&lt;&lt; name;' added inside the main function. The status bar at the bottom indicates 'Ln 12, Col 2', '100%', 'Windows (CRLF)', and 'UTF-8'.</p>	 <p>The screenshot shows a command prompt window titled 'C:\Windows\System32\cmd.exe'. It displays the compilation command 'D:\Codes&gt;c++ test.cpp -o test.exe', the execution command 'D:\Codes&gt;test.exe', and the output 'Ali'.</p>

**Activity:** Write The above code on your computer editor to check the value stored in the variable.

### Example#2:

Declare a variable named “**studentname**”.

```
string studentname;
studentname = “Qaiser”;
```

```
#include<iostream>

using namespace std;

int main()
{

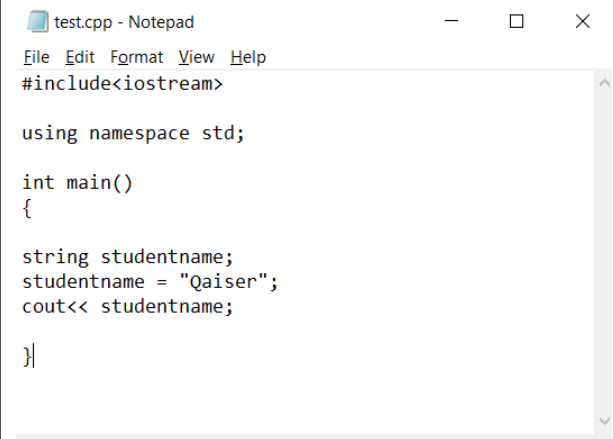
string studentname;
studentname = "Qaiser";

}|
```

**Activity:** Write the above code on your computer editor to declare a string-type variable in your program.

You can check the value stored in this variable by using the “**cout<<**” instruction.

Let's add the **cout** instruction into our program.



```
test.cpp - Notepad
File Edit Format View Help
#include<iostream>

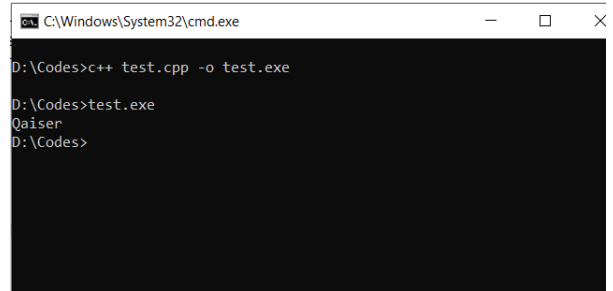
using namespace std;

int main()
{

string studentname;
studentname = "Qaiser";
cout<< studentname;

}|
```

The output of the above code is as follows.



```
C:\Windows\System32\cmd.exe
D:\Codes>c++ test.cpp -o test.exe
D:\Codes>test.exe
Qaiser
D:\Codes>
```

**Activity:** Write The above code on your computer editor to check the value stored in the variable.



## How to output on console?

For this purpose, the C++ language has an instruction that is defined as

```
cout<< "Your Statement";
```

The instruction must be given in this order explicitly and only then the computer would be able to understand it.

Now, Let's use it to print your first line on the computer console.

Example1.

Print "Hello World!" on the computer screen.

```
cout<< "Hello World!";
```

**Task:** Write this instruction within the program structure to print it on the screen.

```
#include<iostream>

using namespace std;

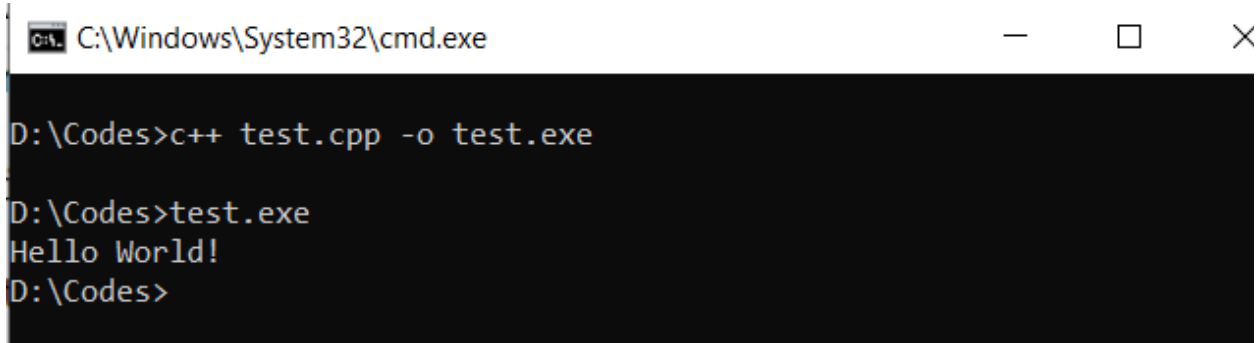
int main()
{

cout<< "Hello world!";

}
```

**Activity:** Write the above code on your computer to print Hello World on your computer screen.

We have executed this program and it generates the following output.



```
C:\Windows\System32\cmd.exe

D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
Hello World!
D:\Codes>
```

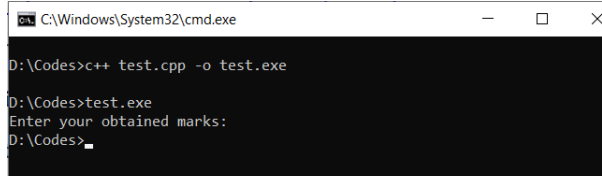
Notice that the program has been successfully executed and “Hello World!” is printed on the console.

## Example#2.

Ask the user to enter his obtained matric marks through the console.

**cout<< “Enter Your obtained Matric Marks: ”;**

**Activity: Write the above code on your computer to print the above-mentioned line on your computer screen.**

Code	The code generates the following output.
<pre>#include&lt;iostream&gt;  using namespace std;  int main() {     cout&lt;&lt; "Enter your obtained marks: "; }</pre>	

Similarly, this instruction is also used to print the value of any variable on the screen.

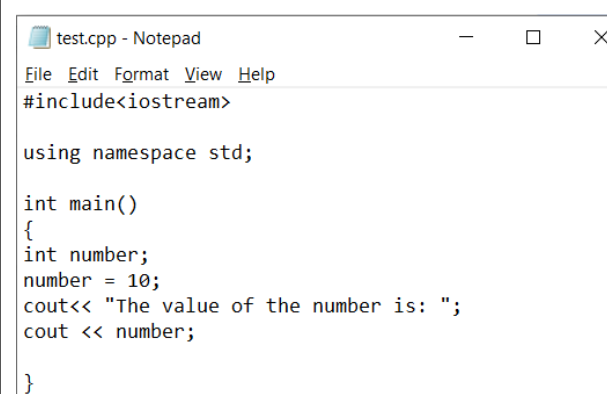
In this format, we shall use the same instruction but replace the “**Enter your obtained matric marks**” with a variable name whose value we want to print on the screen.

For example, See the attached code snippet.	This code generates the following output.
	

**Activity: Write the above code on your computer editor to print the value of the number on your computer screen.**

OR we can do the same things by using the following method.

This code generates the following output.

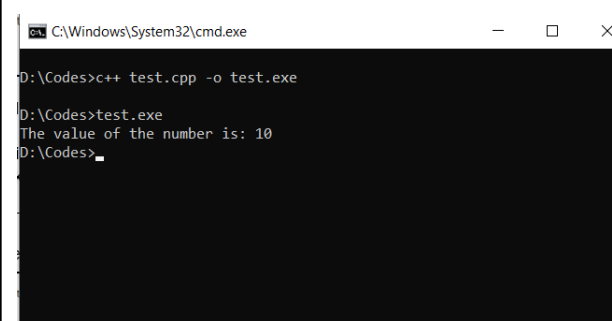


```
test.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{
    int number;
    number = 10;
    cout<< "The value of the number is: ";
    cout << number;

}
```



```
C:\Windows\System32\cmd.exe
D:\Codes>c++ test.cpp -o test.exe
D:\Codes>test.exe
The value of the number is: 10
D:\Codes>
```

**Activity: Write the above code on your computer editor to print the value of the number on your computer screen.**

**endl statement:**

Until now, we have learned about various types of variables and output instruction. However, what if we wanted to move to the next line on the console?

For example

We want to print on screen that

**My name is Ali**

**I have obtained 780 marks**

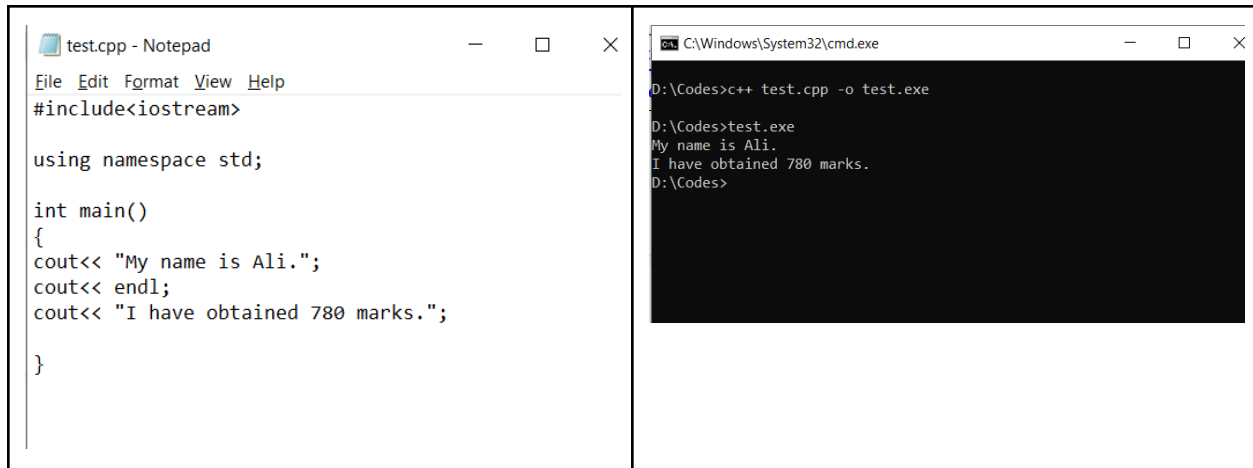
Notice that **Ali** is printed on the first line and **780** is printed on the second line. We can use the “endl” in the “cout<<” instruction to move the cursor to the next adjacent line.

To solve the problem, write the following code using your keyboard.

```
cout<< "My name is Ali.";
cout<< endl;
cout<< "I have obtained 780 marks.";
```

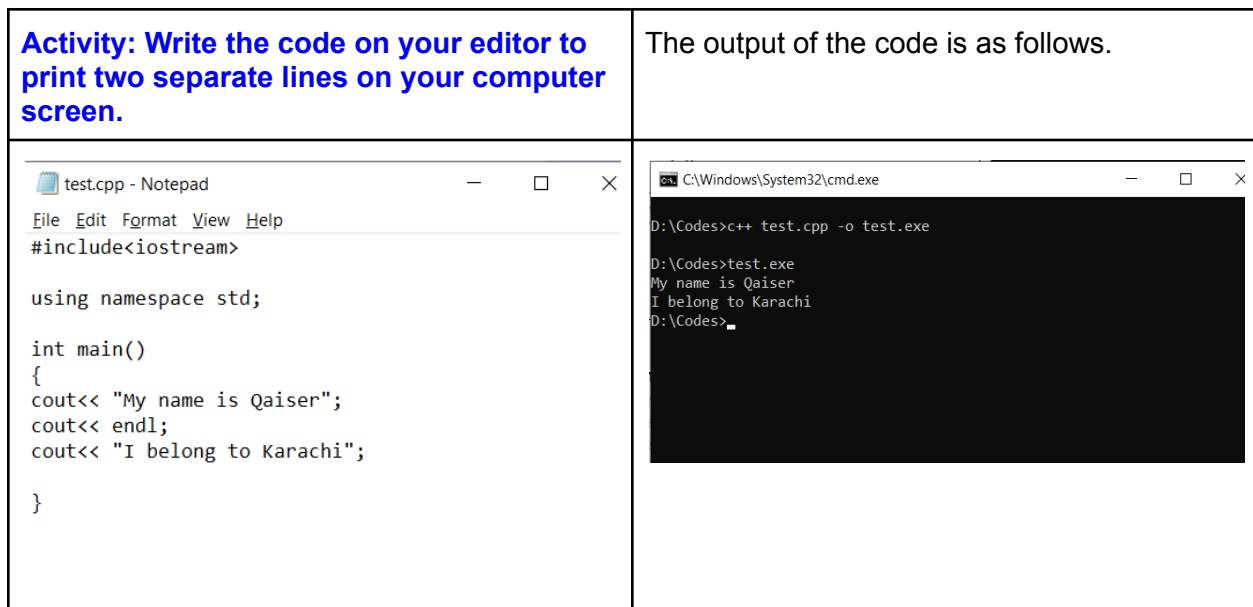
**Activity: Write this code in your editor to print two separate lines on your computer screen.**

The code generates the following output.



### Example 2:

```
cout<< "My name is Qaiser";
cout<< endl;
cout<< "I belong to Karachi";
```



Write a C++ program that prints a box using stars (\*).

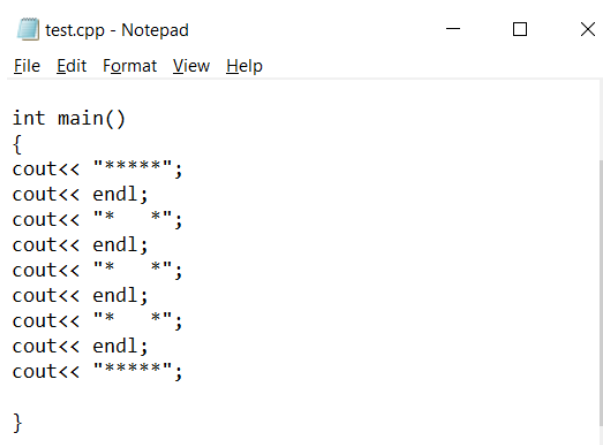
```
*****
*   *
*   *
*   *
*****
```

First Try by yourself.

**Activity:** Try to solve the above question by using the knowledge you have learned to this point.

Don't worry.

The solution is attached below.

Solution:	The output of the code is as follows.
 <pre>test.cpp - Notepad File Edit Format View Help  int main() { cout&lt;&lt; "*****"; cout&lt;&lt; endl; cout&lt;&lt; " *   *"; cout&lt;&lt; endl; cout&lt;&lt; " *   *"; cout&lt;&lt; endl; cout&lt;&lt; " *   *"; cout&lt;&lt; endl; cout&lt;&lt; "*****"; }</pre>	 <pre>C:\Windows\System32\cmd.exe D:\Codes&gt;c++ test.cpp -o test.exe D:\Codes&gt;test.exe *****  *   *  *   *  *   * ***** D:\Codes&gt;</pre>

## How to get input?

For this purpose, the C++ language has an instruction that is defined as **cin>> variable;**

The instruction must be given in this order explicitly and only then the computer would be able to understand it.

Now, Let's use it to get your first variable input from the user.

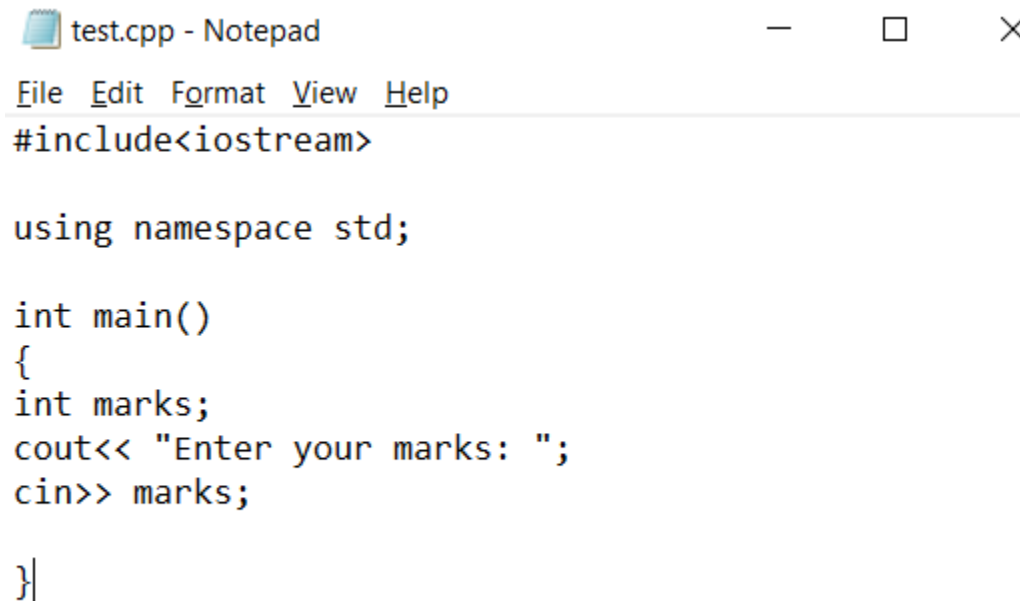
**int marks;**

**cout<< "Enter your marks: ";**

**cin>> marks;**

write the above code in the compiler to take input from the user.

It should look like this.



```
test.cpp - Notepad
File Edit Format View Help
#include<iostream>

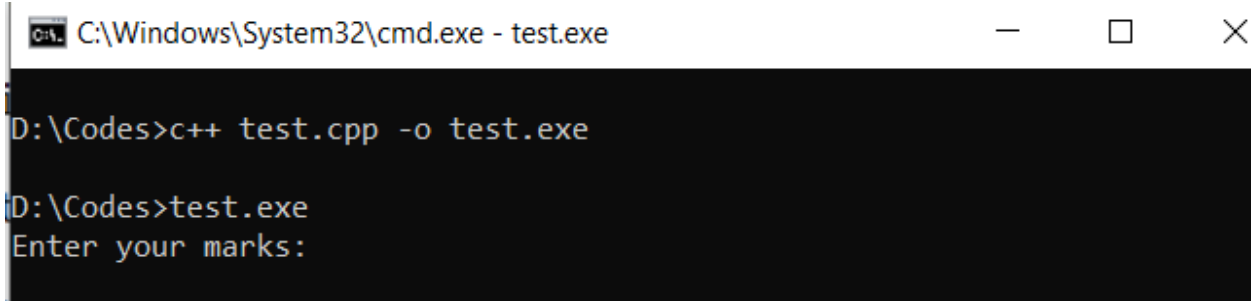
using namespace std;

int main()
{
int marks;
cout<< "Enter your marks: ";
cin>> marks;

}
```

**Activity: Write the above code on your editor to get the value of the “marks” variable from the user.**

When the above code is executed the following output is displayed on the screen.



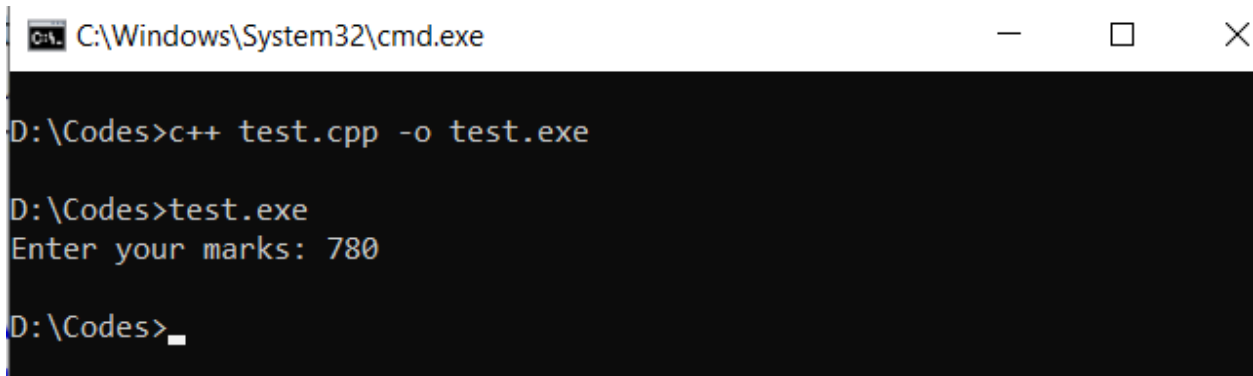
```
C:\Windows\System32\cmd.exe - test.exe

D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
Enter your marks:
```

In this way, the execution of the code stops until the user provides an input (in this case, the user input his marks).

Let's insert "780" using the keyboard.



```
C:\Windows\System32\cmd.exe

D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
Enter your marks: 780

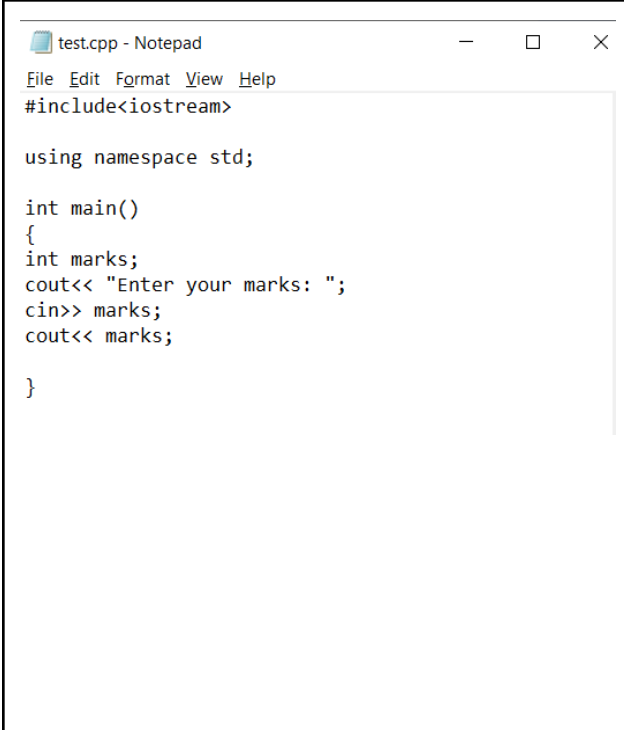
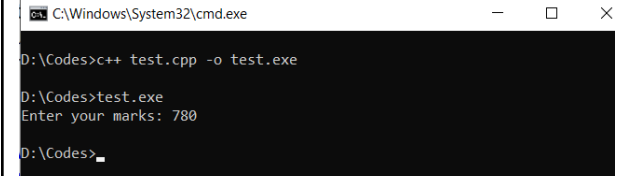
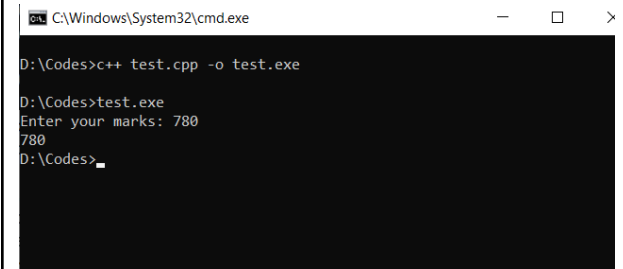
D:\Codes>_
```

Notice that the program has successfully completed its execution when we have given it a number.

Now, the entered value shall be stored in the “marks” variable.

To countercheck, you can use the below statement to check that this variable stores the given value.

**cout<< marks;**

The resulting code is as follows.	The output of the code is as follows.
	<p>-first time the program asks the user to give <b>input</b> (the marks of the student). We have given <b>780</b>.</p>  <p>-Now observe that the variable has stored the given value into the <b>marks</b> variable.</p> 

**Activity:** Write the above code on your computer editor to check the value of the “marks” variable stored in your computer.

## Expressions

Now that we have learned about variables, input, and output instructions, it's time that we learn about manipulating these variables to perform different tasks.

For example:

**Developing a simple expression to add two numbers.**

```
#include<iostream>

using namespace std;

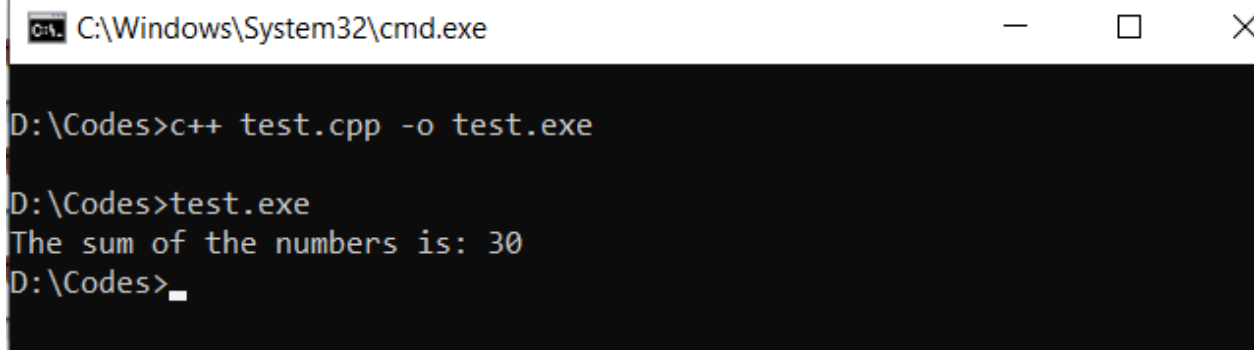
int main()
{
    int number1;
    number1 = 10;
    int number2;
    number2 = 20;
    int sum;
    sum = number1+number2;

    cout<< "The sum of the numbers is: ";
    cout<< sum;

}
```

**Activity: Write the above code in your editor in order to get the answer to the developed expression.**

The output for the above code is as follows.



```
C:\Windows\System32\cmd.exe

D:\Codes>c++ test.cpp -o test.exe

D:\Codes>test.exe
The sum of the numbers is: 30
D:\Codes>
```

Similarly, we can use these variables to perform various mathematical tasks as well.



### Example#2:

Consider the following question.

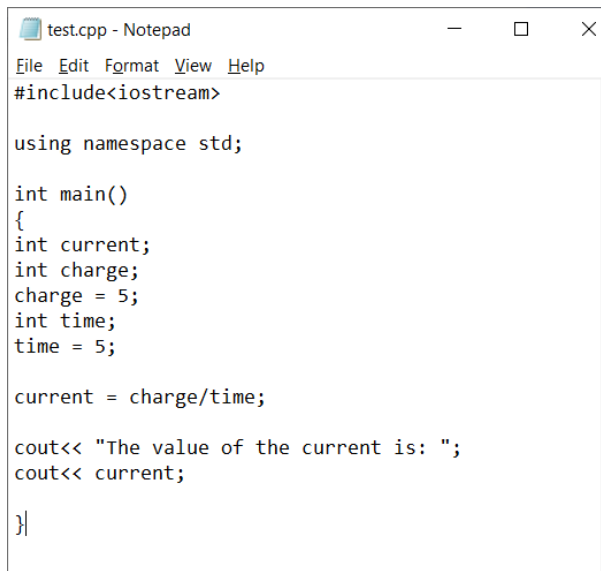
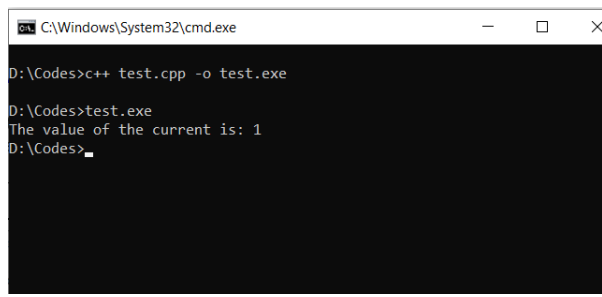
Write a program to calculate current (I) in a wire. The charge (Q) flowing through it in a time (t) of 5 seconds is 5 Coulombs. Print the current (I) on the screen.

Hint:  $I = Q / t$

First, try yourself.

Don't worry.

The solution is attached below. For Now, Just try for yourself.

Solution:	The output for the above code is as follows.
	

**Activity:** Write the above code on your editor to get the answer using the expression.

Now, let's increase the complexity a little bit.

### Example#3

Consider the same question but with the following restrictions.

Write a program that takes charge (Q) and time (t) as input from the user and prints the current (I) on the console.

### solution:

```
test.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{
    float current;

    float charge;
    cout<< "Enter the value of charge: ";
    cin>> charge;

    int time;
    cout<< "Enter the value of time: ";
    cin>> time;

    current = charge/time;

    cout<< "The value of the current is: ";
    cout<< current;

    }|
```

The output of the code is as follows.

First the program asks to enter charge

```
C:\Windows\System32\cmd.exe - test.exe

D:\Codes>c++ test.cpp -o test.exe
D:\Codes>test.exe
Enter the value of charge: _
```

Secondly, the program asks you to enter time

```
C:\Windows\System32\cmd.exe - test.exe

D:\Codes>c++ test.cpp -o test.exe
D:\Codes>test.exe
Enter the value of charge: 12
Enter the value of time:
```

Finally, The output is provided.

```
C:\Windows\System32\cmd.exe

D:\Codes>c++ test.cpp -o test.exe
D:\Codes>test.exe
Enter the value of charge: 12
Enter the value of time: 6
The value of the current is: 2
D:\Codes>_
```

**Activity:** Write the above code on your computer to get the answer to the developed expression.

## Scenario

Assume that Ali is a student who wants to calculate his aggregate for taking admission in UET. we shall use a computer program that would take his obtained marks and after processing, it would tell Ali his aggregate.

Firstly, we shall need to ask Ali about his marks and store those values somewhere so we can calculate the final aggregate at the end.

In programming, there are “nicknames” that are known as “**variables**” that are used to store such values.

So we shall use variables to store the obtained marks of Ali.

Let's name the variable as “**matric**” for storing matric marks.

Let's name the variable as “**inter**” for storing first-year marks.

Let's name the variable as “**ecat**” for storing ecat marks.

Now, as these marks are going to be **integer numbers**, we shall use the integer datatype for storing their value into the variables. And by this far, we know that every line of code in c++ language is ended with a semicolon.

Let's declare the variables.

```
int matric;  
int inter;  
int ecat;
```

Now, ask Ali to input the values for these variables.

```
cout << “Enter your matric marks: ”;  
cin >> matric;  
cout << “Enter your firstyear marks: ”;  
cin >> inter;  
cout << “Enter your ecat marks: ”;  
cin >> ecat;
```

Now, we need to calculate his aggregate.

This is where we shall incorporate the concepts of “**expressions**”.

Now we need to calculate 25% of matric marks, 45% of first-year marks, and 30% of ecat marks to calculate Ali's final aggregate.

We shall have more variables to calculate the respective percentage of each result. Let's declare these variables as well.

```
int matric_percent;  
int inter_percent;  
int ecat_percent;
```

Now, Let's use the expression to calculate the percentage.

```
matric_percent = matric/1100.0*100*0.25;  
inter_percent = inter/550*100*0.45;  
ecat_percent = ecat/400*100*0.30;
```

You are almost there.

We have successfully built logic for your question.

Now, Let's add the three individual percentages to calculate the final aggregate.

```
int total;  
total = matric_percent + inter_percent + ecat_percent;
```

Let's tell Ali about his calculated aggregate.

```
cout<< "Your aggregate is: " << total;
```

**"Congratulations, you have finished your first complete c++ program".**

Now, we shall use all these concepts to perform the tasks that are listed below.

## Program # 1.

Write a c++ program to calculate the aggregate of your obtained marks in the below format.

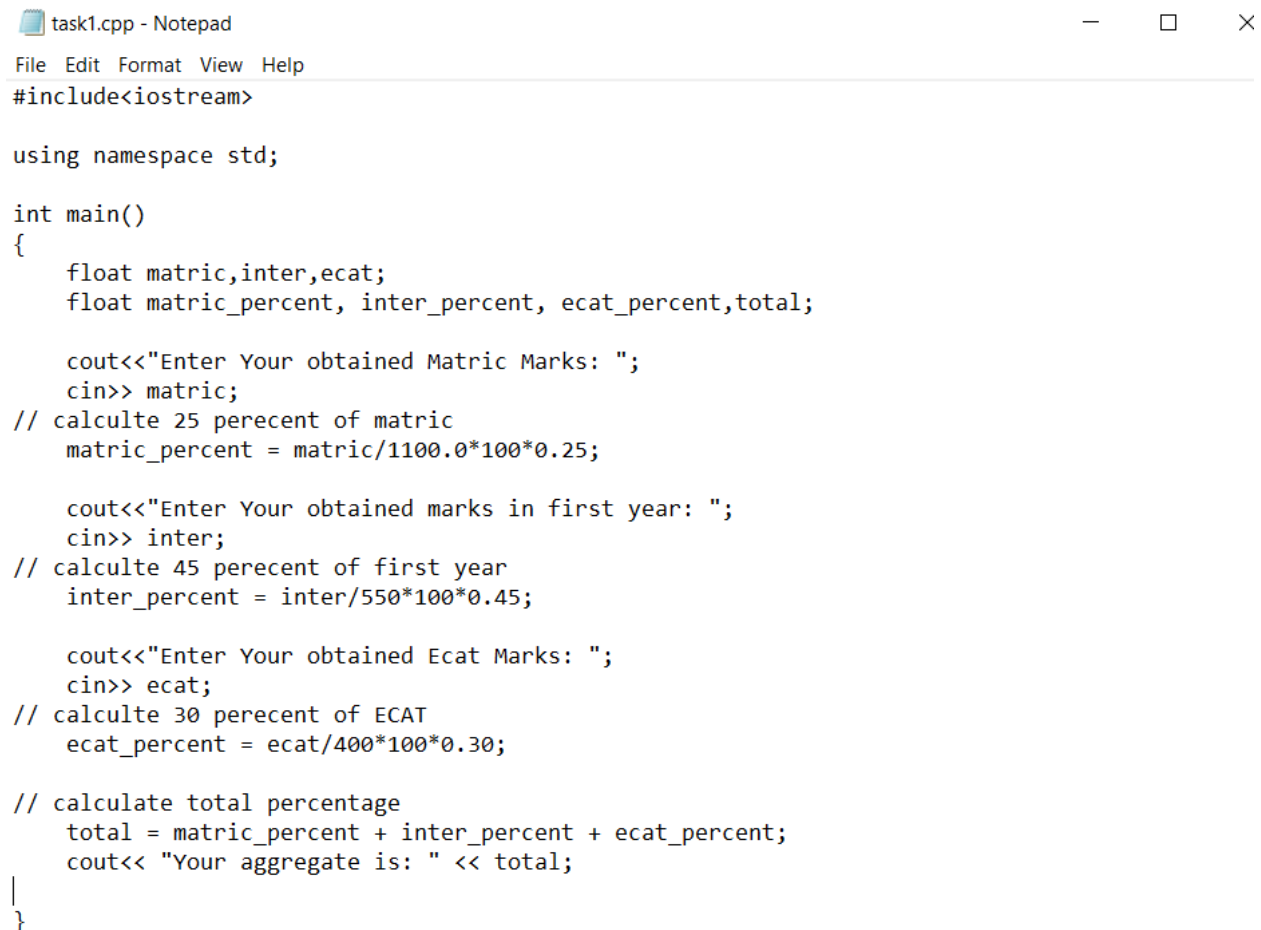
**Enter your matric marks:**

**Enter your First-year marks:**

**Enter your ECAT marks:**

**Your Aggregate is:**

## Solution:



```
task1.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{
    float matric,inter,ecat;
    float matric_percent, inter_percent, ecat_percent,total;

    cout<<"Enter Your obtained Matric Marks: ";
    cin>> matric;
    // calculate 25 perecent of matric
    matric_percent = matric/1100.0*100*0.25;

    cout<<"Enter Your obtained marks in first year: ";
    cin>> inter;
    // calculate 45 perecent of first year
    inter_percent = inter/550*100*0.45;

    cout<<"Enter Your obtained Ecat Marks: ";
    cin>> ecat;
    // calculate 30 perecent of ECAT
    ecat_percent = ecat/400*100*0.30;

    // calculate total percentage
    total = matric_percent + inter_percent + ecat_percent;
    cout<< "Your aggregate is: " << total;
}
```

## Output:

```
C:\Windows\System32\cmd.exe

D:\Codes>c++ task1.cpp -o task1.exe

D:\Codes>task1.exe
Enter Your obtained Matric Marks: 780
Enter Your obtained marks in first year: 320
Enter Your obtained Ecat Marks: 120
Your aggregate is: 52.9091
D:\Codes>
```

## Program # 2A.

Write a c++ program to print the first five multiples of the given number.

For example, if the number is 3.

**Enter the number: 3**

The output should be

**The multiples are: 3 6 9 12 15**

## Solution:

```
task2.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{
    int num ;
    cout<< "Enter the number: ";
    cin>> num;
    // build a logic to print the series
    cout<< "Series is: " << num << " " << num+num << " " << num+num+num << " " << num+num+num+num << " " << num+num+num+num+num << " " << num+num+num+num+num+num;

    return 0;
}
```

## Output:

```
C:\Windows\System32\cmd.exe

D:\Codes>c++ task2.cpp -o task2.exe

D:\Codes>task2.exe
Enter the number: 3
Series is: 3 6 9 12 15
D:\Codes>
```

## Program # 2B.

Write a c++ program to print the first five multiples of two given numbers.

For example, if the input is 3 and 5.

**Enter the first number: 3**

**Enter the second number: 5**

The output should be

**The multiples of the first number are: 3 6 9 12 15**

**The multiples of the second number are: 5 10 15 20 25**

## Solution:

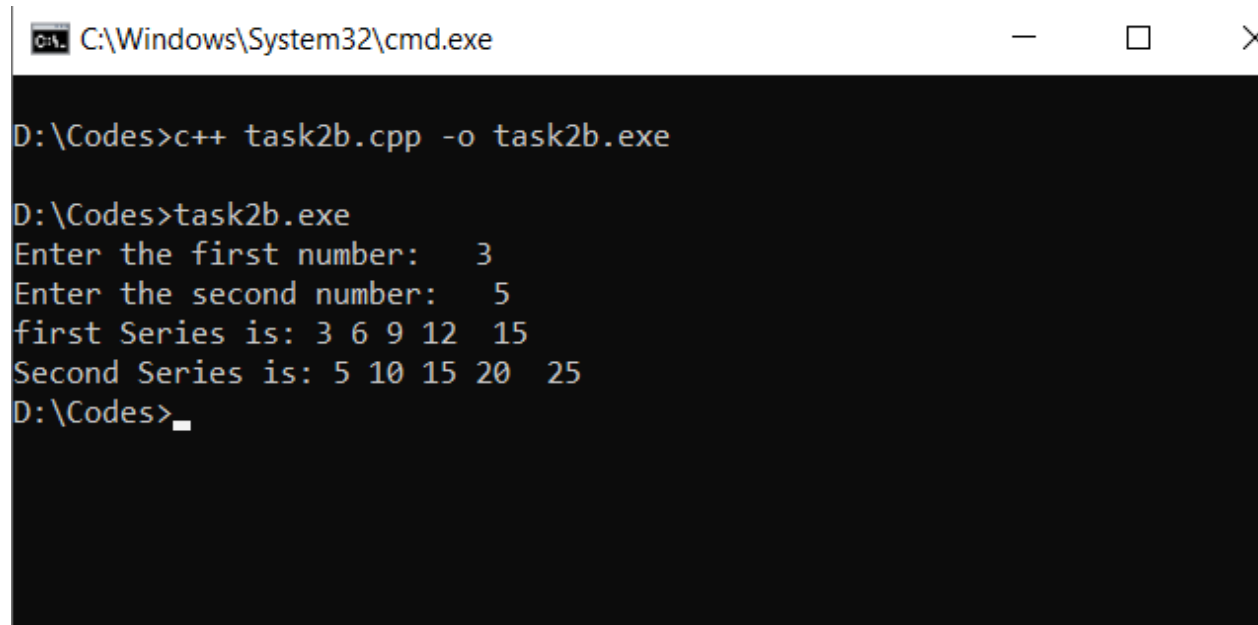
```
*task2b.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{
    int num1, num2 ;
    cout<< "Enter the first number: ";
    cin>> num1;

    cout<< "Enter the second number: ";
    cin>> num2;
    // build the logic
    cout<< "first Series is: ";
    cout << num1 << " " << num1+num1 << " " << num1+num1+num1 << " " << num1+num1+num1+num1 << " " << num1+num1+num1+num1+num1 << " " << num1+num1+num1+num1+num1+num1;
    //build the logic
    cout<< endl << "Second Series is: ";
    cout<< num2 << " " << num2+num2 << " " << num2+num2+num2 << " " << num2+num2+num2+num2 << " " << num2+num2+num2+num2+num2;
}
}
```

## Output:



```
C:\Windows\System32\cmd.exe

D:\Codes>c++ task2b.cpp -o task2b.exe

D:\Codes>task2b.exe
Enter the first number: 3
Enter the second number: 5
first Series is: 3 6 9 12 15
Second Series is: 5 10 15 20 25
D:\Codes>
```

### Program # 2C.

Write a c++ program to print the sum of the first five multiples of two given numbers.

For example, if the input is 3 and 5.

**Enter the first number: 3**

**Enter the second number: 5**

The output should be

**The multiples of the first number are: 3 6 9 12 15**

**The multiples of the second number are: 5 10 15 20 25**

**The sum of the two multiples is: 120**



## Solution:

```
*task2c.cpp - Notepad
File Edit Format View Help
#include<iostream>

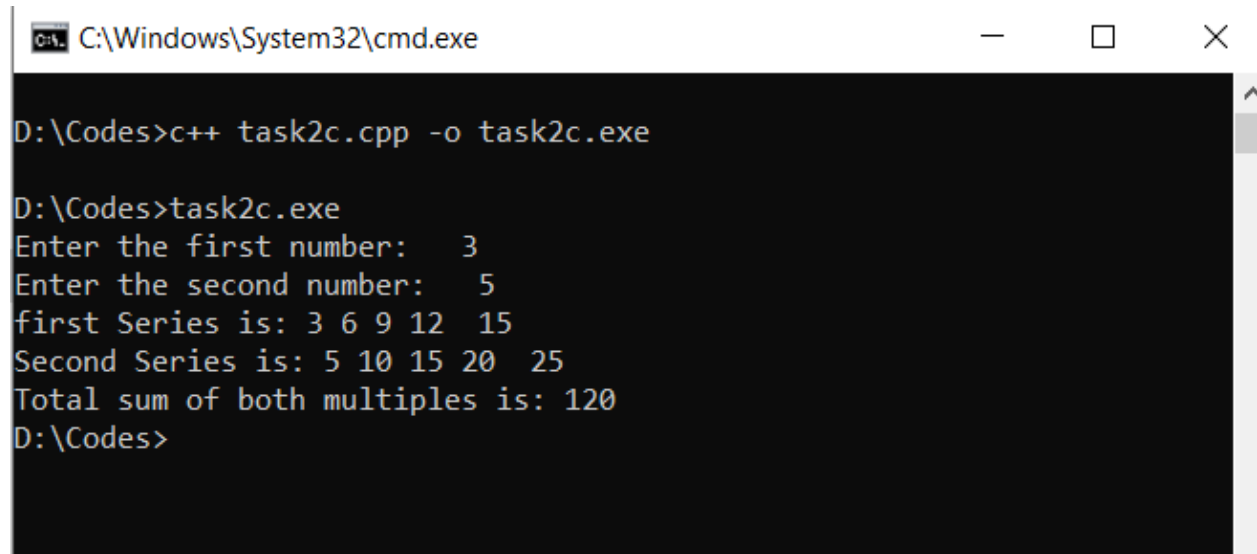
using namespace std;

int main()
{
    int num1, num2, sum1, sum2, total_sum ;
    cout<< "Enter the first number: ";
    cin>> num1;

    cout<< "Enter the second number: ";
    cin>> num2;

    cout<< "first Series is: " ;
    cout<< num1 << " " << num1+num1 << " " << num1+num1+num1 << " " << num1+num1+num1+num1 << " " << num1+num1+num1+num1+num1;
    // building logic for addition of all the numbers of first series
    sum1 = num1 + (num1+num1) + (num1+num1+num1) + (num1+num1+num1+num1) + (num1+num1+num1+num1+num1);
    cout<< endl << "Second Series is: " ;
    cout<< num2 << " " << num2+num2 << " " << num2+num2+num2 << " " << num2+num2+num2+num2 << " " << num2+num2+num2+num2+num2;
    // building logic for addition of all the number of second series
    sum2 = num2 + (num2+num2) + (num2+num2+num2) + (num2+num2+num2+num2) + (num2+num2+num2+num2+num2);
    // total of both series
    total_sum = sum1 + sum2;
    cout<< endl << "Total sum of both multiples is: " << total_sum;
}
```

## Output:



```
C:\Windows\System32\cmd.exe

D:\Codes>c++ task2c.cpp -o task2c.exe

D:\Codes>task2c.exe
Enter the first number: 3
Enter the second number: 5
first Series is: 3 6 9 12 15
Second Series is: 5 10 15 20 25
Total sum of both multiples is: 120
D:\Codes>
```

## Program # 3.

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Write a program that takes two numbers from the user and prints the Fibonacci series from that point up to 5 terms.

## Solution:

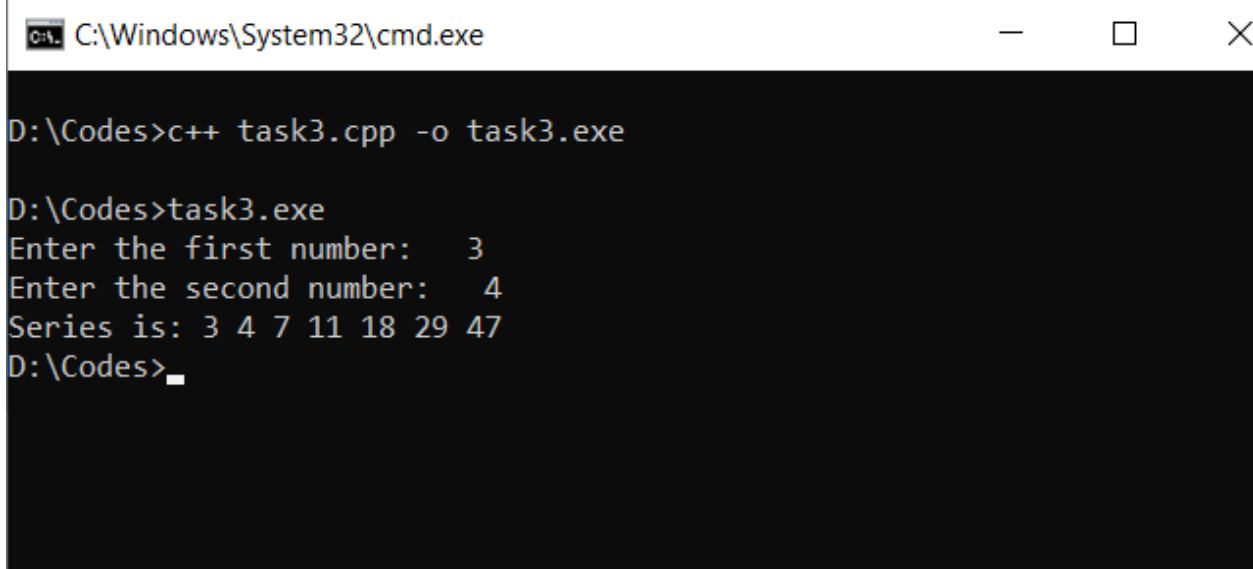
```
task3.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{
    int num1, num2 ;
    cout<< "Enter the first number:  ";
    cin>> num1;

    cout<< "Enter the second number:  ";
    cin>> num2;
    //build the logic
    cout<< "Series is: " << num1 << " " << num2 << " " << num1+num2 << " " << num2 + (num1+num2) << " "
    << (num1+num2)+ (num2 + (num1+num2)) << " " << (num2 + (num1+num2)) + ((num1+num2)+ (num2 + (num1+num2)))
    << " " << ((num1+num2)+ (num2 + (num1+num2))) + ((num2 + (num1+num2)) + ((num1+num2)+ (num2 + (num1+num2)))) ;
}
}
```

## Output:



```
C:\Windows\System32\cmd.exe

D:\Codes>c++ task3.cpp -o task3.exe

D:\Codes>task3.exe
Enter the first number: 3
Enter the second number: 4
Series is: 3 4 7 11 18 29 47
D:\Codes>
```

## Program # 4.

The sequence of numbers (1, 2, 3, ... , 100) is arithmetic and when we are looking for the sum of a sequence, we call it a series. Thanks to Gauss, there is a special formula we can use to find the sum of a series:

$$S = \frac{n(n+1)}{2}$$

Write a program that takes input from the user and prints the sum of consecutive numbers to the input value.

For example, if the input is

**Enter the number: 100**

The output should be:

**The sum of all the numbers is: 5050**

**Solution:**



\*task4.cpp - Notepad



File Edit Format View Help

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int num, sum;
```

```
    cout<< "Enter the number:  ";
```

```
    cin>> num;
```

```
// build the logic and apply the formula
```

```
    sum = num*(num+1)/2;
```

```
    cout<< "Total Sum is:  "<< sum;
```

```
}
```

**Output:**

```
C:\Windows\System32\cmd.exe

D:\Codes>c++ task4.cpp -o task4.exe

D:\Codes>task4.exe
Enter the number: 100
Total Sum is: 5050
D:\Codes>
```

## Program #5.

Take two numbers as input and find the sum between these two numbers.

Input n1= 2

Input n2=8

Processing answer= $2+3+4+5+6+7+8 = 35$

### Hint:

Formula:  $(n / 2)(\text{first number} + \text{last number}) = \text{sum}$ , where n is the number of integers between two numbers and  $n = \text{last number} - (\text{first number} - 1)$

### Solution:

```
*task5.cpp - Notepad
File Edit Format View Help
#include<iostream>

using namespace std;

int main()
{
    int num1, num2, sum;
    float total_numbers;

    cout<<"Enter First Number: ";
    cin>> num1;

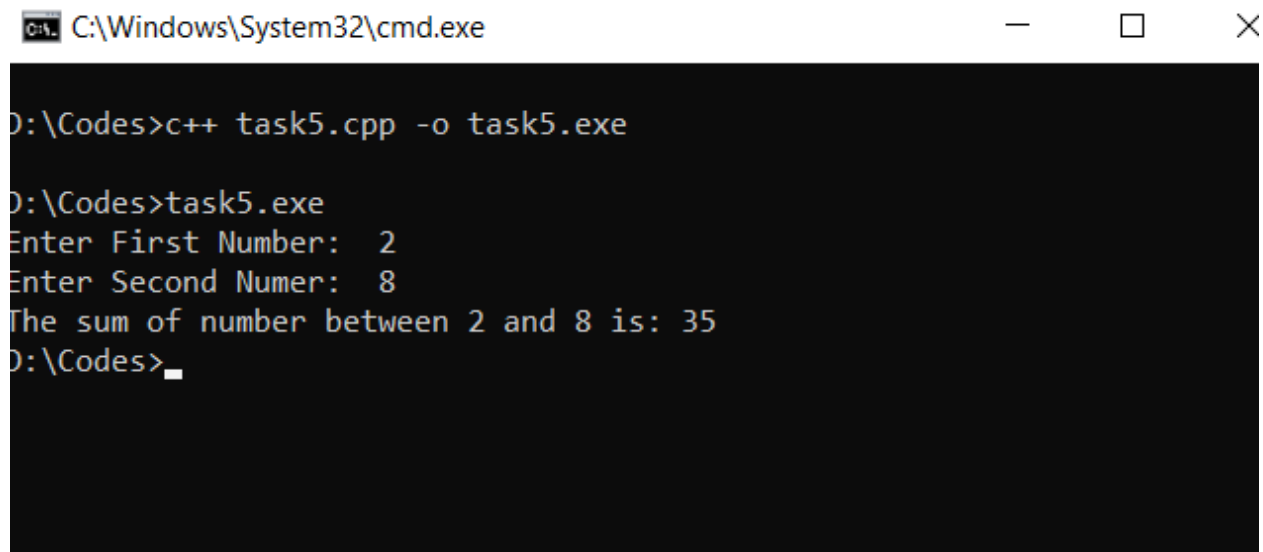
    cout<<"Enter Second Number: ";
    cin>> num2;

    total_numbers = num2-(num1-1);
    // build the logic and apply the formula
    sum = (total_numbers/2)*(num1+num2);

    cout<<"The sum of number between " <<num1<< " and "<< num2<<" is: "<<sum;

}
```

Output:



```
C:\Windows\System32\cmd.exe

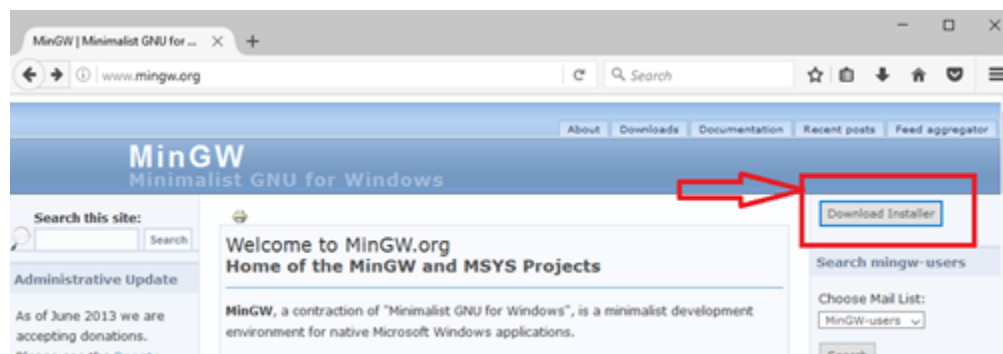
D:\Codes>c++ task5.cpp -o task5.exe

D:\Codes>task5.exe
Enter First Number: 2
Enter Second Numer: 8
The sum of number between 2 and 8 is: 35
D:\Codes>_
```

## Mingw Installation and Setting Up Environment Variables

Installation of C++ Compiler

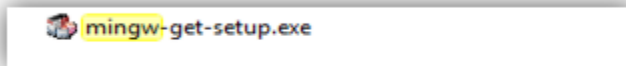
- open <http://www.mingw.org/>
- Click Download Button as you can see in the picture below.



- When you click the download button on mingw website. The following page will open in your browser (from the SourceForge.net website).



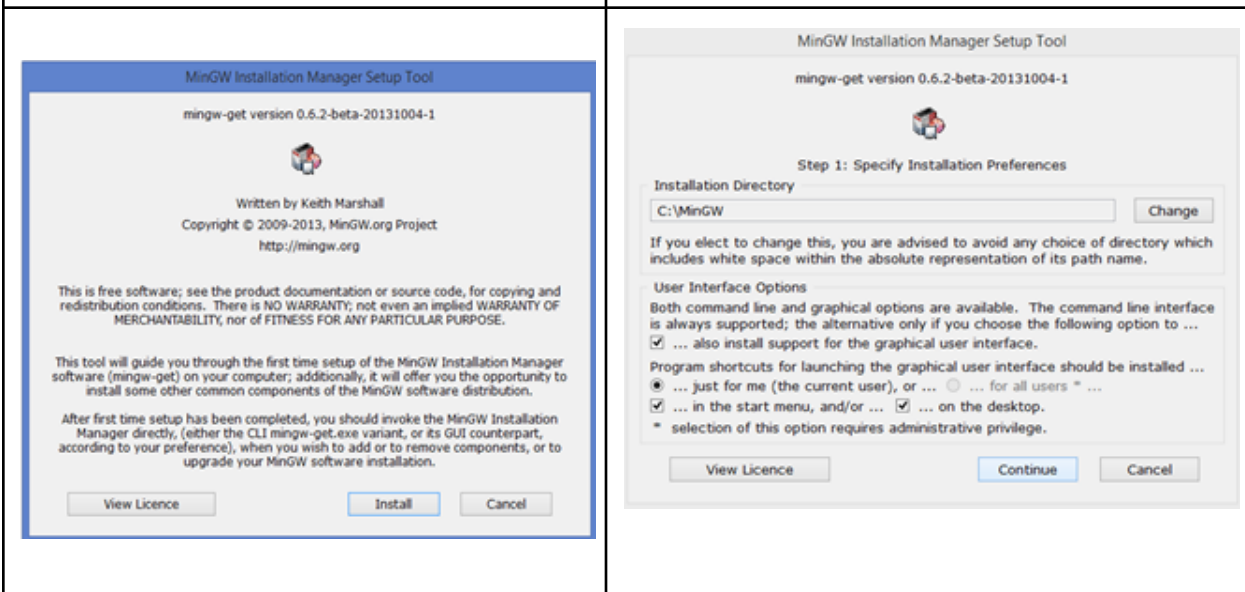
The following exe file will be downloaded with



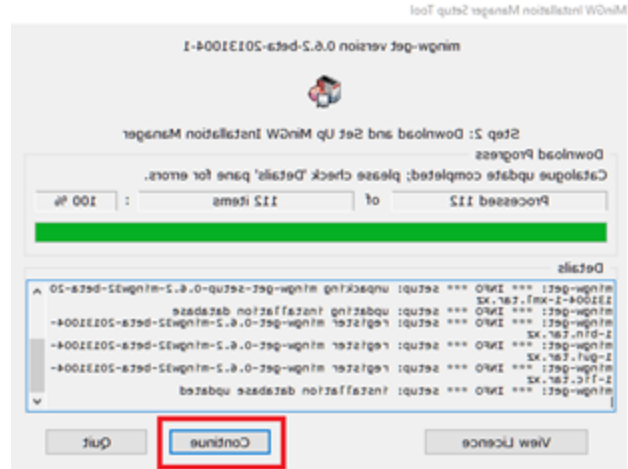
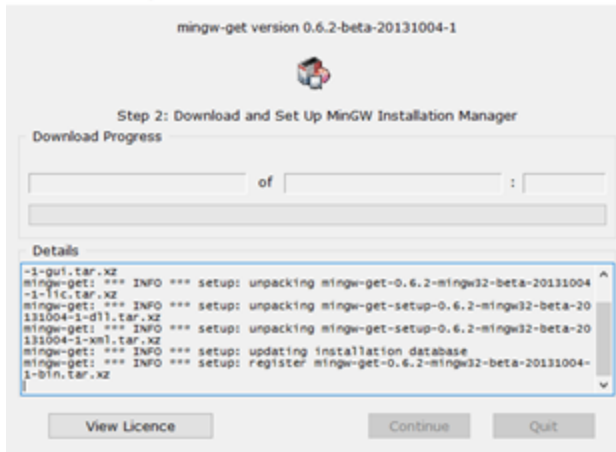
Perform the following steps

- Click **mingw-get-setup.exe**
- Click install

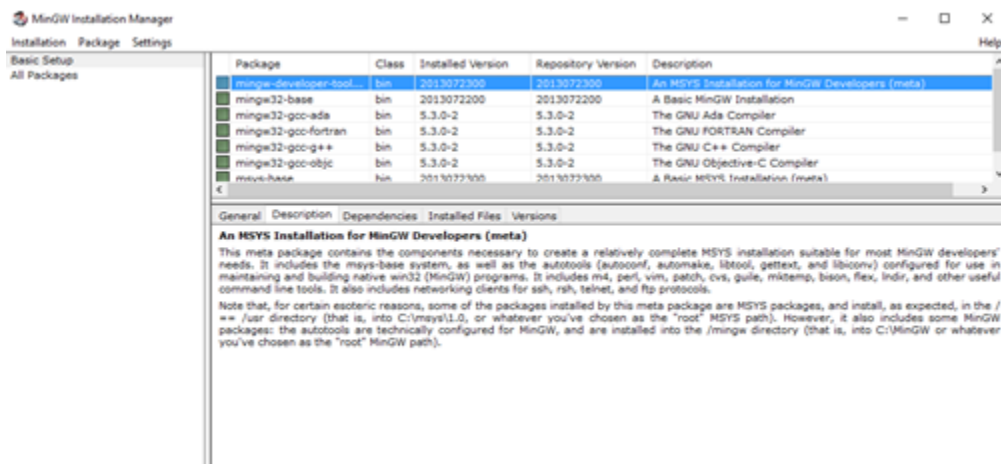
- Installation directory is C:\MinGW
- Click continue



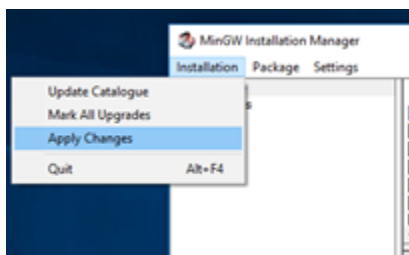
Now click continue



- The following pop-up window will appear. Please make sure that you selected all the check-boxes. e.g. mingw32-base, mingw32-gcc=g++, msys-base and so on [A1] [RS2] .

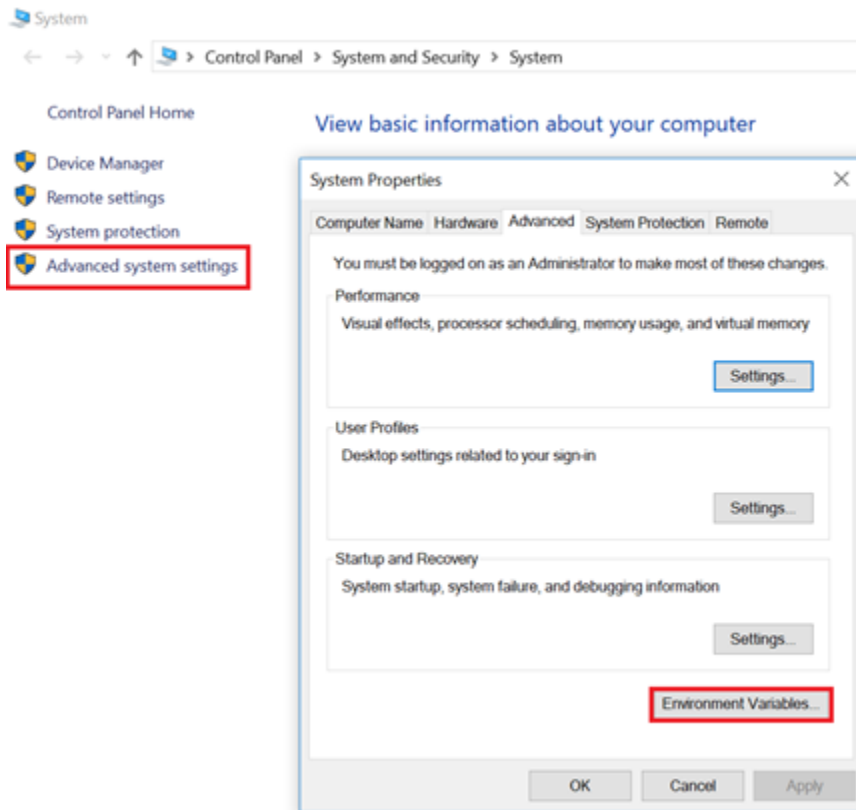


- Click on Installation > Apply Changes as shown in the picture below.
- Wait for the process to complete. Once you see a successful installation message close the window. Click **Close**.
- Now we will set the environment variable to use gcc [A1] and g++ [A2] [RS3] commands from the terminal.



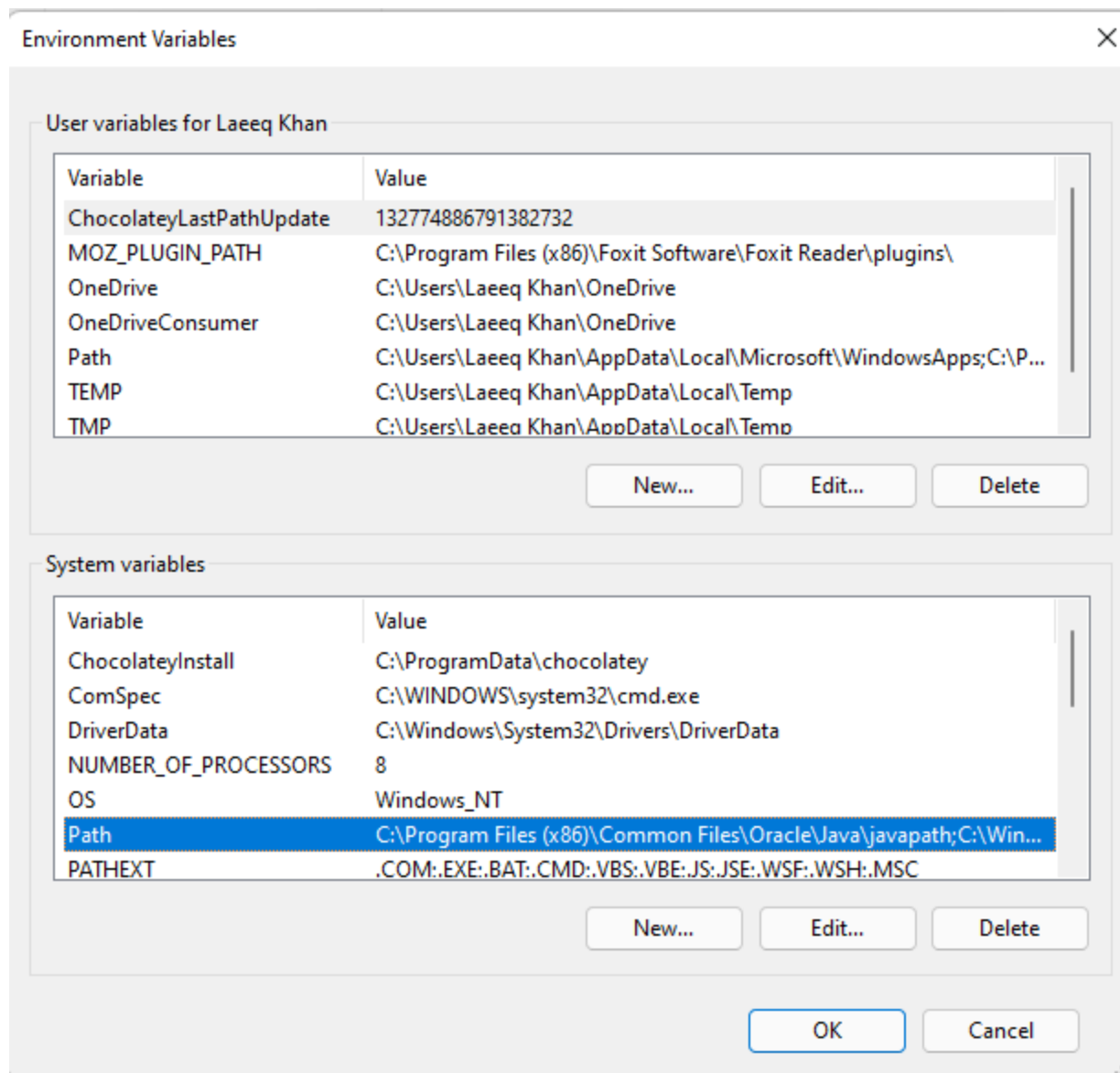
## Setting up PATH Variable for Windows

- Once Mingw is installed, we want to execute C++ programs from command prompt; we have to add its bin directory to the PATH variable.
- Go to installation Directory and copy the path of bin folder, e.g. “C:\MinGW\bin”
- Go to “Control Panel\System and Security\System”, Click “Advanced System Settings” and then “Environment Variables”.

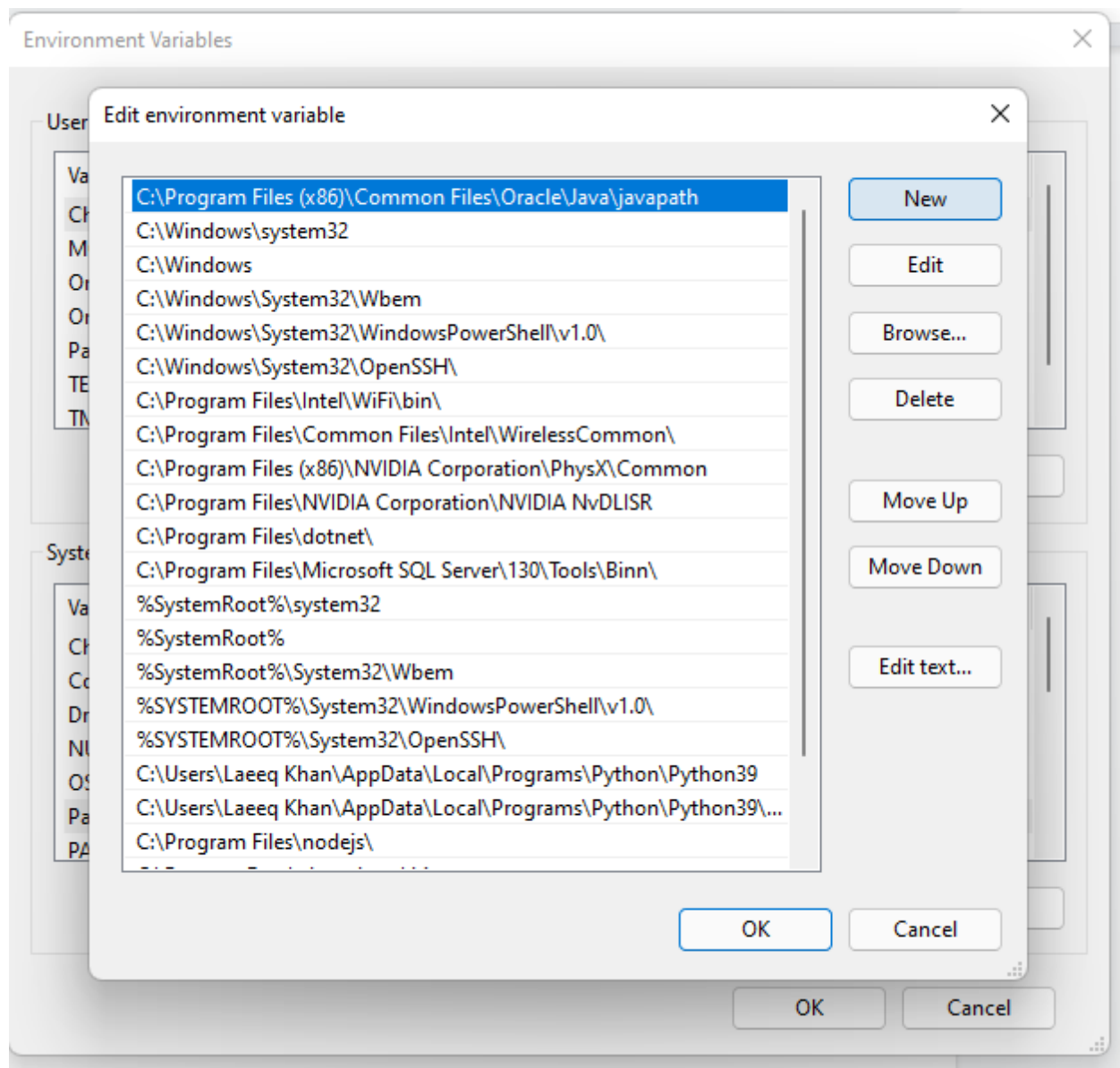


- Find the Variable Name “Path” in User or System (if you are administrator) Variables and click on Edit.





- Select Path and Click on Edit Button



- Click on “New” and paste the “bin” folder path.
- Press OK on all opened popup windows.
- Close all CMDs and Open cmd again and write “g++”[A1] in it, press enter/return key. G++ is successfully installed and you can exit the cmd.
- In case you get the message “‘g++’ [A2] is not recognized as an internal or external command”, follow steps correctly for installation and configuration of Path variable.
- That’s all for installing C++ compiler in Windows. We are ready now to explore coding features of C++ Programming.

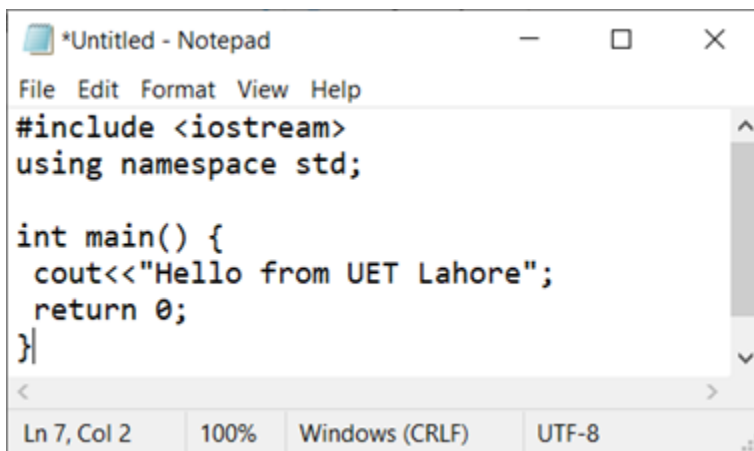
**Note:** Now as you have installed C++ Compiler on your computer, you are ready to write and run your first program on your machines. Repl account also gives you the freedom to write and execute your programs in your browser without installation of C++ compiler on your machines. We highly recommend that you should complete your work on your laptops and use Repl for online submission of your tasks only.

### First C++ Program:

- To write the first C++ program, open the notepad on the computer.
- To open the notepad, press windows key + R from your keyboard.
- Write a notepad on the opened screen as shown in the figure.
- Press Ok.

Notepad editor will appear on screen, where you can write your first program.

- Write the given code in the opened window as shown in the figure. We will see the details in a while, let's first execute our first program.

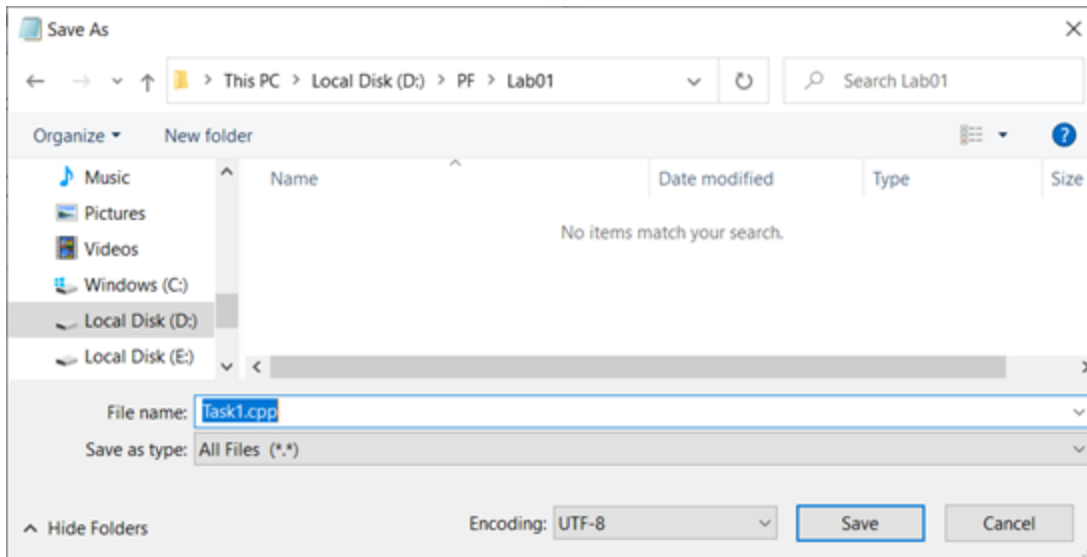


```
*Untitled - Notepad
File Edit Format View Help
#include <iostream>
using namespace std;

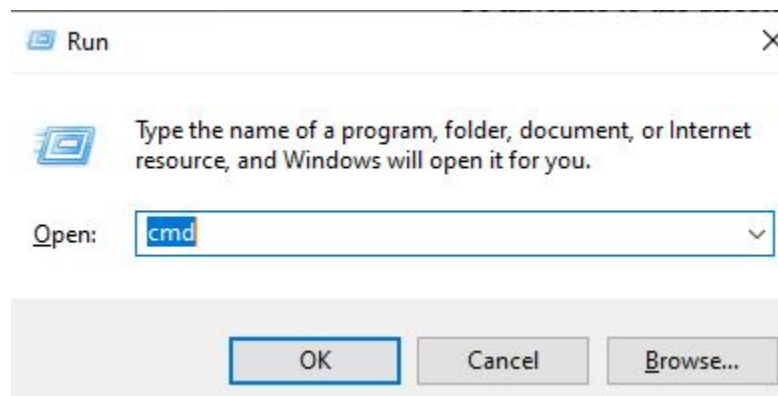
int main() {
    cout<<"Hello from UET Lahore";
    return 0;
}

Ln 7, Col 2    100%    Windows (CRLF)    UTF-8
```

- Now that we have completed code, we need to save our program. Press Ctrl + S from the keyboard.
- Navigate to the folder where you want to save the written code.
- In this case, we will save the file in directory D>> PF >> Lab01.
- Change the Save as Type to "All files".
- Name this file as Task1.cpp



- Click on the save button.
- Now our program is saved onto the computer. We are ready to execute it.
- To execute your program, open the command prompt.
- To open the command prompt, press Windows key + R, write cmd [L1] in the text field and press enter.



- To navigate to the directory in the cmd, we are using the given command.
- First three commands help us to navigate to our desired directory.

To compile the program, we will use c++ command. If you see an error on c++ command that c++ command is not recognized then you have to revisit the step for setting of environment variables.

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Samyan>D:

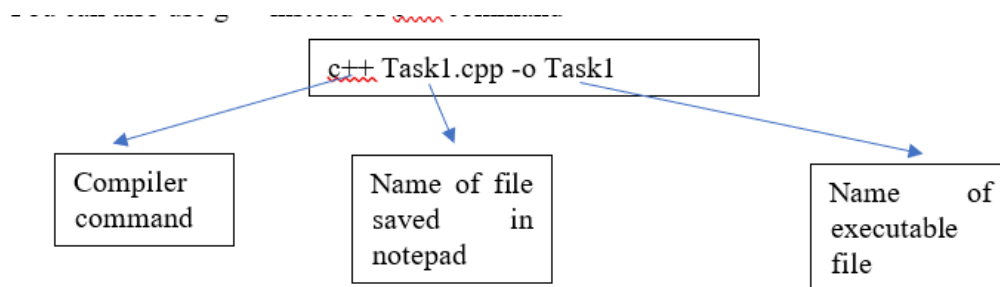
D:\>cd PF

D:\PF>cd Lab01

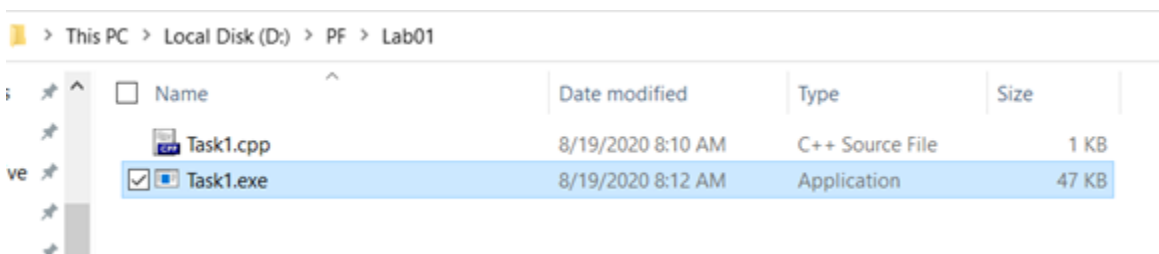
D:\PF\Lab01>g++ Task1.cpp -o Task1

D:\PF\Lab01>Task1
Hello from UET Lahore
D:\PF\Lab01>
```

- To compile the program, following command will be executed:
- You can also use g++ instead of c++ command



- You can set name of executable file to any name, we used Task1 as file name



- As a result, you will see the executable file with the .exe extension in the same folder.
- Next write the name of the executable file to like task1.exe or whatever you name chose.

```
C:\Windows\system32\cmd.exe

C:\Users\Laeq-Khan\Desktop>c++ Task1.cpp -o task1
C:\Users\Laeq-Khan\Desktop>task1.exe
```

- Code you can see is a template for writing any C++ program except line 4. Whenever you want to write a C++ program, you have to write this skeleton code. Without this skeleton, your C++ basics are incomplete and you cannot run your program.
- Details on the code are available in the handouts of the first week.

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

int main() {
    cout<<"Hello from UET Lahore";
    return 0;
}
```

### Output

```
C:\Users\Laeeg-Khan\Desktop>c++ Task1.cpp -o task1
C:\Users\Laeeg-Khan\Desktop>task1.exe
Hello from UET Lahore
C:\Users\Laeeg-Khan\Desktop>
```

**Congratulations! You have learnt all the basic essential concepts of the mingw compiler and executed your first c++ program by using the cmd.**

