



<ul style="list-style-type: none"><li>• Formatted Input and output functions</li><li>• output text</li><li>• Output numeric values</li><li>• Output Arithmetic expression</li><li>• un Formatted Input and output functions</li></ul>	السابع - الثامن
---	-----------------

## Formatted Input and output functions

Learn: What are the **console input, output related methods, operations in C++**? In this topic we learn how to use input / output operations in C++ language.

Submitted by [Amit Shukla](#), on June 17, 2017

Console input / output function take input from standard input devices and compute and give output to standard output device.

Generally, keyboard is standard input device and monitor is standard output device.

**In case of C++ it uses streams to perform input and output operations in standard input output devices (keyboard and monitor). A stream is an object which can either insert or extract the character from it.**

The standard C++ library is **iostream** and standard input / output functions in C++ are:

1. **cin**
2. **cout**

**There are mainly two types of consol I/O operations form:**

1. Unformatted consol input output
2. Formatted consol input output

### 1) Unformatted consol input output operations



These input / output operations are in unformatted mode. The following are operations of unformatted consol input / output operations:

## A) void get()

It is a method of `cin` object used to input a single character from keyboard. But its main property is that it allows wide spaces and newline character.

### Syntax:

```
char c=cin.get();
```

### Example:

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

int main()
{
    char c=cin.get();
    cout<<c<<endl;

    return 0;
}
```

## Output

```
I
I
```

## B) void put()

It is a method of `cout` object and it is used to print the specified character on the screen or monitor.

### Syntax:

```
cout.put(variable / character);
```



### Example:

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

int main()
{
    char c=cin.get();
    cout.put(c); //Here it prints the value of variable c;
    cout.put('c'); //Here it prints the character 'c';

    return 0;
}
```

### Output

```
I
Ic
```

## C) getline(char \*buffer,int size)

This is a method of `cin` object and it is used to input a string with multiple spaces.

### Syntax:

```
char x[30];

cin.getline(x,30);
```

### Example:

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

int main()
{
    cout<<"Enter name :";
```



```
char c[10];  
cin.getline(c,10); //It takes 10 charcters as input;  
cout<<c<<endl;  
  
return 0;  
}
```

## Output

```
Enter name :Divyanshu  
Divyanshu
```

## D) write(char \* buffer, int n)

It is a method of `cout` object. This method is used to read n character from buffer variable.

### Syntax:

```
cout.write(x,2);
```

### Example:

```
#include<iostream>  
using namespace std;  
  
int main()  
{  
    cout<<"Enter name : ";  
    char c[10];  
    cin.getline(c,10); //It takes 10 charcters as input;  
    cout.write(c,9); //It reads only 9 character from  
    buffer c;  
  
    return 0;  
}
```

## Output



```
Enter name : Divyanshux  
Divyanshu
```

## E) cin

It is the method to take input any variable / character / string.

### Syntax:

```
cin>>variable / character / String / ;
```

### Example:

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

int main()
{
    int num;
    char ch;
    string str;

    cout<<"Enter Number"<<endl;
    cin>>num; //Inputs a variable;
    cout<<"Enter Character"<<endl;
    cin>>ch; //Inputs a character;
    cout<<"Enter String"<<endl;
    cin>>str; //Inputs a string;

    return 0;
}
```

## Output

```
Enter Number
07
Enter Character
h
Enter String
```



Deepak

## F) cout

This method is used to print variable / string / character.

### Syntax:

```
cout<< variable / charcter / string;
```

### Example:

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

int main()
{
    int num=100;
    char ch='X';
    string str="Deepak";

    cout<<"Number is "<<num<<endl; //Prints value of
variable;
    cout<<"Character is "<<ch<<endl; //Prints character;
    cout<<"String is "<<str<<endl; //Prints string;

    return 0;
}
```

## Output

```
Number is 100
Character is X
String is Deepak
```

## 2) Formatted console input output operations



In formatted console input output operations we use the following functions to make output in perfect alignment. In industrial programming all the output should be perfectly formatted due to this reason C++ provides many functions to convert any file into perfect aligned format. These functions are available in header file `<iomanip>`. `iomanip` refers to input output manipulations.

## A) width(n)

This function is used to set the width of the output.

### Syntax:

```
cout<<setw(int n);
```

### Example:

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

int main()
{
    int x=10;
    cout<<setw(20)<<variable;

    return 0;
}
```

## Output

```
10
```

## B) fill(char)

This function is used to fill specified character at unused space.

### Syntax:

```
cout<<setfill('character')<<variable;
```



### Example:

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

int main()
{
    int x=10;
    cout<<setw(20);
    cout<<setfill('#')<<x;

    return 0;
}
```

### Output

```
#####10
```

### D) precision(n)

This method is used for setting floating point of the output.

#### Syntax:

```
cout<<setprecision('int n')<<variable;
```

### Example:

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

int main()
{
    float x=10.12345;
    cout<<setprecision(5)<<x;

    return 0;
}
```





## Output

```
10.123
```

### E) setflag(arg 1, arg,2)

This function is used for setting format flags for output.

**Syntax:**

```
setiosflags(argument 1, argument 2);
```

### F) unsetflag(arg 2)

This function is used to reset set flags for output.

**Syntax:**

```
resetiosflags(argument 2);
```

### G) setbase(arg)

This function is used to set basefield of the flag.

**Syntax:**

```
setbase(argument);
```