C++ Console I/O Function

C++ *Console I/O* function

C++ language provides us console input/output functions. As the name says, the console input/output functions allow us to -

- Read the input from the keyboard, entered by the user at the console.
- Display the output to the user at the console.

Note : These input and output values could be of *any primitive data type*.

There are two kinds of console input/output functions :

No.	Functions
1	*Formatted input/output functions.*
2	*Unformatted input/output functions.*

- ## *Unformatted input/output functions*

Unformatted console input/output functions are used for performing input/output operations at console and the resulting data is left unformatted and untransformed i.e. it is left in its raw and original form.

- In C++, we can *read the input* entered by a user at console using an object **cin** of *istream* class and through this object we can access the functions of *istream* class, such as - ***get(char *), get(void) and getline()***.

- In C++, we can *write the output* at console using an object **cout** of *ostream* class and through this object we can access the functions of *ostream* class, such as - ***put(), write()***.

Some of the most important formatted console input/output functions are -

Functions	Description
get(char *)	Reads a *single* character from the user at the console and assigns it to the char array in its argument, but needs an **Enter** key to be pressed at the end..
get()	Reads a *single* character from the user at the console, *and returns it*.
getline(char* arr, int size)	Reads a line of characters, entered by the user at the console which ends with a newline character or until the size of .
put(char ch)	Writes a *single* character at the console.
write(char *arr, int num)	Writes a number of characters in a char array to the console.

```cpp
#include<iostream>
using namespace std;
int main()
{
    char ch;
    cout<<"Enter Character=";
    //cin.get(ch);
    ch=cin.get();//reading
    cout.put(ch);//writing
    return 0;
}
```

```cpp
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    fstream new_file;
    new_file.open("new_file_write.txt",ios::in);
    if(!new_file)
    {
        cout<<"No such file";
    }
    else
    {
        char ch[50];//showing the content
        while(!new_file.eof())
        {
            new_file.getline(ch,50);
            new_file.write(ch,50);
            new_file>>ch;
            cout<<ch ;

        }
        new_file.close();
        return 0;
    }
}
```

- ## *Formatted input/output functions*

Formatted console input/output functions are used for performing input/output operations at console and the resulting data is formatted and transformed. For more on formatted input/output functions, please read **formatted input/output functions**.

Functions	Description
width(int width)	Using this function, we can specify the width of a value to be displayed in the output at the console.
fill(char ch)	Using this function, we can fill the unused white spaces in a value(*to be printed at the console*), with a character of our choice.
setf(arg1, arg2)	Using this function, we can set the *flags*, which allow us to display a value in a particular format.
unsetf(char ch)	Using this function, we could clear the flag specified fixed by the function setf().
precision(int num_of_digts)	Using this function, we can specify the number of digits(*num_of_digits*) to the right of decimal, to be printed in the output.

In C++, we can *read the input* entered by a user at console using an object **cin** of *istream* class and we can *write the output* at console using an object **cout** of *ostream* class. Through the **cin** and **cout** objects, we can access the formatted I/O functions.

```cpp
#include<iostream>
using namespace std;
int main()
{
    //1
    cout<<100<<endl;
    //2
    cout.width(10);
    cout<<100<<endl;
    //3
    cout.fill('x');
    cout.width(20);
    cout<<std::right<<100<<endl;
    return 0;
}
```

```cpp
using namespace std;
int main()
{
    //cout<<34;
    //cout.setf(ios::hex, ios::basefield);
    //cout<<34;
    cout.setf(ios::oct, ios::basefield);
    //cout<<34;
    //cout.setf(ios::showbase);
    //cout<<34;
    //cout.unsetf(ios::showbase);
    //cout<<34;
    cout.setf(ios::fixed, ios::floatfield);
    cout.precision(20);
    cout<<34.1;

}
```