ShreeSoft INFORMATICS

Address: Tirumala Plaza, Upendra Nagar, Cidco, Nashik.

Join us on Telegram:
https://t.me/ShreeSoftinformatics

Join us on WhatsApp:
https://chat.whatsapp.com/BrLM0ZoWMbpA8nXi9Mrocl

Contact: **+91 8308341531**

Formatted I/O in C++

The C++ programming language provides the several built-in functions to display the output in formatted form. These built-in functions are available in the header file iomanip.h and ios class of header file iostream.h.

C++ helps you to format the I/O operations like determining the number of digits to be displayed after the decimal point, specifying number base etc.

There are two ways to do so:
1. Using the ios class or various ios member functions.
2. Using the special functions called manipulators defined in iomanip.h.

Formatted IO using ios class memebers:

- The ios class contains several member functions that are used to perform formatted IO operations.
- The ios class also contains few format flags used to format the output.
- It has format flags like showpos, showbase, oct, hex, etc.
- The format flags are used by the function setf( ).

The following table provides the details of the functions of ios class used to perform formatted IO in C++.

Function	Description
width(int)	Used to set the width in number of character spaces for the immediate outputdata.
fill(char)	Used to fill the blank spaces in output with given character.
precision(int)	Used to set the number of the decimal point to a float value.
setf(format flags)	Used to set various flags for formatting output like showbase, showpos, oct,hex, etc.
unsetf(format flags)	Used to clear the format flag setting.

All the above functions are called using the built-in object cout.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
	int x=78;
	cout<<"Example for formatted IO";

	cout<<endl<< "Default:"<<endl;
	cout<<123;

	cout<<endl<<"width(5):";
	cout.width(-5);
	cout<<endl<<123;

	cout<<endl<<"width(5) and fill('*'): " << endl;
	cout.width(5);
	cout.fill('#');
	cout<<123<<endl;


	cout.precision(2);
	cout << "precision(2) ---> " << 123.4567890 << endl;
	cout << "precision(2) ---> " << 9.876543210 << endl;

	cout << "setf(showpos): " << endl;
	cout.setf(ios::showpos);
	cout<<123<<endl;

	cout << "unsetf(showpos): " << endl;
	cout.unsetf(ios::showpos);
	cout<<123<< endl;

	cout<<endl<<"HexDecimal:";
	cout<<hex<<x;
	cout<<endl<<"Octal:";
	cout<<oct<<x;
}
```

Formatted IO using manipulators

The iomanip.h header file contains several special functions that are used to perform formmated IO operations.

The following table provides the details of the special manipulator functions used to perform formatted IO in C++.

Function	Description
setw(int)	Used to set the width in number of characters for the immediate outputdata.
setfill(char)	Used to fill the blank spaces in output with given character.
setprecision(int)	Used to set the number of digits of precision.
setbase(int)	Used to set the number base.
setiosflags(format flags)	Used to set the format flag.
resetiosflags(format flags)	Used to clear the format flag.

The **iomanip.h** also contains the following format flags using in formatted IO in C++.

Flag	Description
endl	Used to move the cursor position to a newline.
ends	Used to print a blank space (null character).
dec	Used to set the decimal flag.
oct	Used to set the octal flag.
hex	Used to set the hexadecimal flag.
left	Used to set the left alignment flag.
right	Used to set the right alignment flag.
showbase	Used to set the showbase flag.
noshowbase	Used to set the noshowbase flag.
showpos	Used to set the showpos flag.
noshowpos	Used to set the noshowpos flag.
showpoit	Used to set the showpoit flag.
noshowpoint	Used to set the noshowpoint flag.

Example:

```
#include<iostream>
#include <iomanip>
using namespace std;
void line() {
        cout<<endl<< "-----------------------------------------";
}
```

```
int main()
{
cout << "Example for formatted IO";
line();
cout<<endl<<"setw(10): ";
cout<<endl<<setw(10)<< 99;
line();

cout<<endl<<"setw(10) and setfill('*'): ";
cout<<setw(10)<< setfill('*')<<99;
line();

cout<<endl<< "setprecision(5): ";
cout<<setprecision(5)<<123.4567890;
line();

cout<<endl<<"showpos: ";
cout << showpos << 999;
line();
cout<<endl<< "hex: ";
cout<<hex <<100;
line();

cout<<endl<<"hex and showbase: ";
cout<<showbase<<hex<<100<;
line();
}
```