

## IO Management

### Writing to the stream:

```
cout << X;
```

In this statement we have used the operator << as an insertion operator. This operator takes two arguments; left hand operand is “cout” which is an object of ostream class, and the right-hand side is an operand whose value is to be written/printed on screen. That is the operator << writes the value of the right-hand operand into the left-hand operand.

We can even use the operator “<<” to chain the output request as:

```
cout << “enter the value “<< endl;
```

endl is a manipulator that has the effect of newline.

### Reading from the stream

```
cin >> X;
```

In this statement we use the “>>” extraction operator the operator takes the istream as its left-hand operand and an object as its right-hand operand. i.e. it reads from the left-hand (“cin”) istream operand and stores the value read into the right-hand operand. >> returns its left hand operand as a result. We can chain the input request to have a combined effect to read two or multiple values as:

```
cin >> x >> y >> z >> .....
```

### example: use of insertion and extraction operator

```
#include<iostream.h>
```

```
void main()
```

```
{
```

```
    int int_var;
```

```
    float float_var;
```

```
    char char_var;
```

```
    cout <<”enter an integer value “<<”.”;
```

```
    cin >> int_var;
```

```

    cout <<"enter a float variable"<<":.";
    cin >> float_var;
    cout <<"enter a character value" <<":.";
    cin >> char_var;
    cout <<"You have entered "<<int_var<<"\t"<<float_var<<"\t"<<char_var<<endl;
}

```

### Single character Input and output functions

getchar(), and putchar(char) are built in 'C' style of function. To make use of them the header file "stdio.h" must be included.

The getchar function return a single character entered through the standard input device (keyboard). The function take no argument and its syntax is :

```
char myChar;
```

```
myChar = getchar();
```

putchar(char): This function is used to display a single character on the standard display device that is the monitor. This function takes the character variable as an argument and returns nothing. The syntax is given below:

```
putchar(myChar);
```

### Example of input and output of a single character:

```
#include <iostream.h>
```

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
Void main()
```

```
{
```

```
    char myChar;
```

```
    cout<<"input a character \t:\t";
```

```
    myChar = getchar()
```

```
    cout<<"\nYou have inputted \t:\t"<<myChar;
}
```

### **String input and output function**

We use the gets() and puts(string) function to transfer the string between computer and standard input/output device. We must include the header file “**stdio.h**” in the source program to make use of gets() and puts() functions.

gets() function need any argument of string or character array, this function read the string from standard input (keyboard) device. The syntax is :

```
char myStrin[20];
```

```
gets(myString);    //this will return a string of size less or equal to 20 characters into myString.
```

### **Displaying the string:**

The function puts () is used to display the string on to the standard output device (that is monitor). This function requires an argument of type string. The syntax and usage is:

```
puts(myString);    //will display the string myString on the display.
```

Example:

```
#include<iostream.h>
```

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
    char myString[10],myName[10];
```

```
    cout <<"Input your name :\t";
```

```
    gets(myName);
```

```
    cout<<"\nYour name is \t:";
```

```
    puts(myName);
```

```
}
```

## I/O Manipulators

Manipulators are used to change certain characteristics of input/output. They change format flag and value of stream. To format the input / output. C++ supports following special functions as i/o manipulators:

|                   |   |
|-------------------|---|
| ENDL              | Generate carriage return or line feed character   |
| SETBASE()         | Used to change base of one numeric value into another base. The common base converters are: dec, hex, oct |
| Setw(int w)       | Used to set width of a variable   |
| Setfill(char )    | Fill the unused field of value  |
| Setprecision(int) | Used to control the number of decimal digits after the decimal point                                      |
| Ends              | Used to insert a null terminating character at the end of the string                                      |
| Flush()           | Used to flush(clear)the output stream   |
|                   |   |

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
#include<iomanip.h>
```

```
void main()
```

```
{
```

```
    int num;
```

```
    clrscr();
```

```
    cout <<"enter a number ";
```

```
    cin>>num;
```

```
    cout<<"the number \t:"<<num<<" in octal is \t : "<<oct<<num<<endl; // now num is in octal
```

```
    cout<<"the number \t:"<<num<<"in hexa is \t:" <<hex<<num<<endl; // num converted oct to hex
```

```
}
```



