Input /Output Streams

https://www.youtube.com/watch?v=EejEQCgxIKM&list=PLoSDia_xPJ5qFeeU_cQLBgfJiJdaZJBkk

Features Of Iostream.h

The standard header file input and output stream (iostream.h) contains a set of small and specific general purpose functions for handling input and output data. The I/O stream is a sequence of following characters written for the screen display or read from the keyboard. The standard input and output operations in C++ are normally performed by using the I/O stream as cin for input and cout for output.

Various stream related header files

C++ compilers	**Header Files To Be Included**
Borland C++ *Turbo C++* *Microsoft Visual C++* *Zortech C++* *UNIX C++ (AT&T)*	iostream . h, ios .h, iomanip . h iostream . h, ios .h, iomanip . h ios . h, istream . h, ostream . h and streamb . h stream . h stream . h

Various stream related header files

- Ios.h
- Istream
- Ostream
- IOstream

ios.h

The standard header file input and output stream (iostream.h) contains a set of small and specific general purpose functions for handling input and output data. The I/O stream is a sequence of following characters written for the screen display or read from the keyboard. The standard input and output operations in C++ are normally performed by using the I/O stream as cin for input and cout for output.

istream.h

The istream consists of input functions to read a streams of characters from the keyboard.

ostream.h

The ostream consists of output functions to write a character onto the screen.

iostream.h

The iostream supports both input/output stream of functions to read a stream of characters form the keyboard and to display a stream of objects onto the video screen.

iostream.h

C++ does not have built-in input and output functions for handling input and output of data from the outside world, but it supports different kind of stream related header files for providing these facilities. The stream library is a hierarchy of classes. The streambuf class is the basis of all streams. It defines the basic characteristics of buffers that hold characters for input and output. The ios class is derived from streambuf. It defines the basic formatting and error control capabilities used in streambuf. The ios is a virtual base class for the classes istream (input stream) and ostream (output stream). The iostream (input/output stream) class is derived from both istream and ostream.

Keyboard and Screen I/O

- ***(a) Cout*** The cout is used to display an object onto the standard device, normally the video screen. The insertion operator (the double less the sign <<) is used along with the cout stream.

  The general syntax of the cout stream is,
- *Cout << variable 1 << variable 2 <<.....<< variable n;*

- ***(b) Cin*** The cin is used to read a number, a character or a string of characters from a standard input device, normally the keyboard. The extraction operator (the double greater then sign >>) is use along with
  the cin operator.

The general syntax of the cin is

cin >> variable 1 >> variable 2 >> …variable n;

Manipulator functions

Manipulator functions

Manipulator functions are special stream functions that change certain characteristics of the input and output. They change the format flags and values for a stream. The main advantage of using manipulator functions is that they facilitate that formatting of input and output streams.

Predefined manipulators

Following are the standard manipulators normally used in the stream classes:

- endl
- setbase
- setw
- setfill
- setprecision
- ends
- ws

(a) Endl the endl is an output manipulator to generate a carriage return or line feed character. The endl may be used several times in a C++ statement.

For example,

(1)

```
cout << “ a “ << endl << “b” << endl;
```

(2)

```
cout << “ a = “ << a << endl;
cout << “ b = “ << b << endl;
```

(b) Setbase() The setbase() manipulator is used to convert the base of one numeric value into another base. Following are the common base converters in C++.

dec - decimal base (base = 10)

hex - hexadecimal base (base = 16)

oct - octal base (base = 8)

PROGRAM

A program to show the base of a numeric value of a variable using hex, oct and dec manipulator functions.

```
    / / using dec, hex, oct manipulator

    #include <iostream.h>
    int main()
{
      int   value;
      cout  << “ Enter number’ << endl;
      cin >> value;
      cout << “ Decimal base = “ << dec << value << endl;
      cout << “ Hexadecimal base = “ << hex << value << endl;
      cout << “ Octal base = “ << oct << value << endl;
   }
```

Output of the above program

```
   Enter number
   10
   Decimal base = 10
   Hexadecimal base = a
   Octal base = 12
```

PROGRAM
A program to show the base of a numeric value of a variable using setbase manipulator function.

```
/ /using setbase manipulator

#include <iostream.h>
#include <iomanip.h>
void main (void)
{
    int    value
    cout << “ Enter number” << endl;
    cin >> value;
    cout << “ decimal base = “ << setbase (10)
    cout << value << endl;
    cout << “ hexadecimal base = “ << setbase (16);
    cout << value << endl;
    cout << “ Octal base = “ << setbase (8) << value << endl;
}
```

(c) Setw ()

The setw ( ) stands for the set width. The setw () manipulator is used to specify the minimum number of character positions on the output field a variable will consume.

The general format of the setw manipulator function is

setw(int w)

Which changes the field width to w, but only for the next insertion. The default field width is 0.

For example,

```
cout << setw (1) << a << endl;
cout << setw (10) << a << endl;
```

PROGRAM

A program to display the data variables using setw manipulator functions.

```
 / /using setw manipulator
 #include <iostream.h>
 #include <iomanip.h>
int main()
 {
     int  a,b;
     a = 200;
     b = 300;
    cout << setw (5) << a << setw (5) << b << endl;
    cout << setw (6) << a << setw (6) << b << endl;
    cout << setw (7) << a << setw (7) << b << endl;
    cout << setw (8) << a << setw (8) << b << endl;
 }
```

Output of the above program

```
  200      300
   200      300
    200      300
     200      300
```

(d) Setfill() The setfill ( ) manipulator function is used to specify a different character to fill the unused field width of the value.

The general syntax of the setfill ( ) manipulator is

setfill(char f)

which changes the fill character to f. The default fill character is a space.

For example,

setfill (‘ . ’) ; / / fill a dot (.) character

setfill (‘ * ’) / / fill a asterisk (*) character

PROGRAM

A program to illustrate how a character is filled in filled in the unused field width of the value of the data variable.

```
    / /using setfill manipulator
    #include <iostream.h>
    #include <iomanip.h>
    int main()
    {
       int   a,b;
       a = 200;
       b = 300;
       cout << setfill ( ‘ * ’ );
       cout << setw (5) << a << setw (5) << b << endl;
       cout << setw (6) << a << setw (6) << b << endl;
Return 0;
}
```

Output of the above program

```
   **200**300
   ***200***300
```

(e) Setprecision() The setprecision ( ) is used to control the number of digits of an output stream display of a floating point value. The setprecision ( ) manipulator prototype is defined in the header file <iomanip.h>.

The general syntax of the setprecision manipulator is

Setprecision (int p)

Which sets the precision for floating point insertions to p. The default precision is 6

PROGRAM

A program to use the setprecision manipulator function while displaying a floating point value onto the screen.

```
 / /using setprecision manipulator
 #include <iostream.h>
 #include <iomanip.h>
int main()
 {
    float  a,b,c;
    a = 5;
    b = 3;
    c = a/b;
    cout << setprecision (1) << c << endl;
    cout << setprecision (2) << c << endl;
    cout << setprecision (3) << c << endl;
    cout << setprecision (4) << c << endl;
    cout << setprecision (5) << c << endl;
}
```

Output of the program

```
1.7
1.67
1.667
1.6667
1.66667
```

(f) Ends The ends is a manipulator used to attach a null terminating character (‘\0’) at the end of a string. The ends manipulator takes no argument whenever it is invoked. This causes a null character to the output.

(g) Ws The manipulator function ws stands for white space. It is used to ignore the leading white space that precedes the first field.

PROGRAM

A program to show how a null character is inserted using ends manipulator while displaying a string onto the screen.

```
    / /using ends manipulator

    #include <iostream.h>
    #include <iomanip.h>
    int main ( )

 {
    int number = 231;
    cout << “ number = “ << number << ends;
    cout << endl;
 }
```

Output of the above program

```
    “ number = 123 “
```

```
/ /using ws manipulator

        #include <iostream.h>
        #include <iomanip.h>
        Void main ( )
        {
            char name [100];
            cout << " enter a line of text \n";
            cin >> ws;
            cin >> name;
            cout  << " typed text is = " << name << endl;
       }
```

Output of the program

```
        enter a line of text
        this is a text
        typed text is = this
```

(b) Flush The flush member function is used to cause the stream associated with the output to be completed emptied.

For input on the screen, this is not necessary as all output is flushed automatically. However, in the case of a disk file begin copied to another, it has to flush the output buffer prior to rewinding the output file for continued use. The function flush ( ) does not have anything to do with flushing the input buffer.

```
/ /using flush member function
#include <iostream.h>
#include <iomanip.h>
void main ( )
{
    cout << “ My name is Computer \n”;
    cout << “ many greetings to you \n”;
    cout . flush ( ) ;
}
```