UNIT-6

MANAGING CONSOLE I/O OPERATIONS

Managing Console I/O Operations

6.1 Input and Output Streams

6.2 C++ Stream Classes

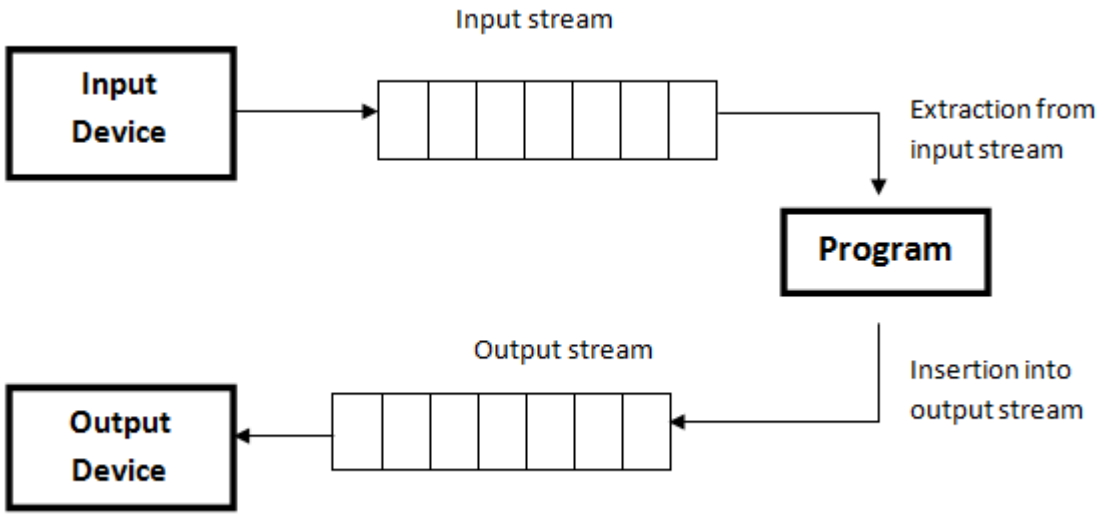
6.3 Unformatted and formatted I/O Operations

6.4 Formatting with Manipulators

VPMP POLYTECHNIC

6.1 Input and Output Streams

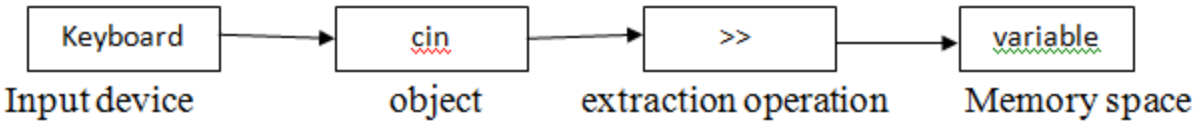
A stream is a sequence of bytes or character.



VP
ЭНИС

Input Streams

- The cin is default connected to keyboard
- We can use cin to receive the input from keyboard

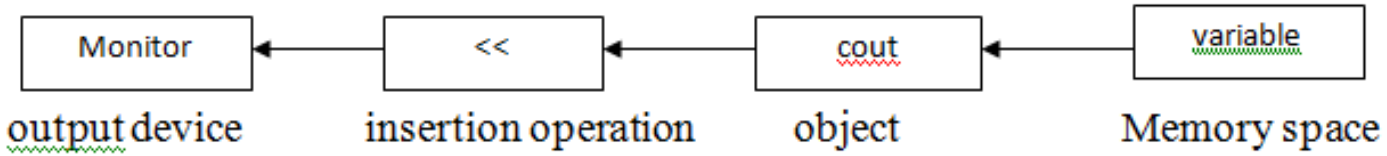


cin stream

VPM TECHNIC

OUTPUT STREAMS

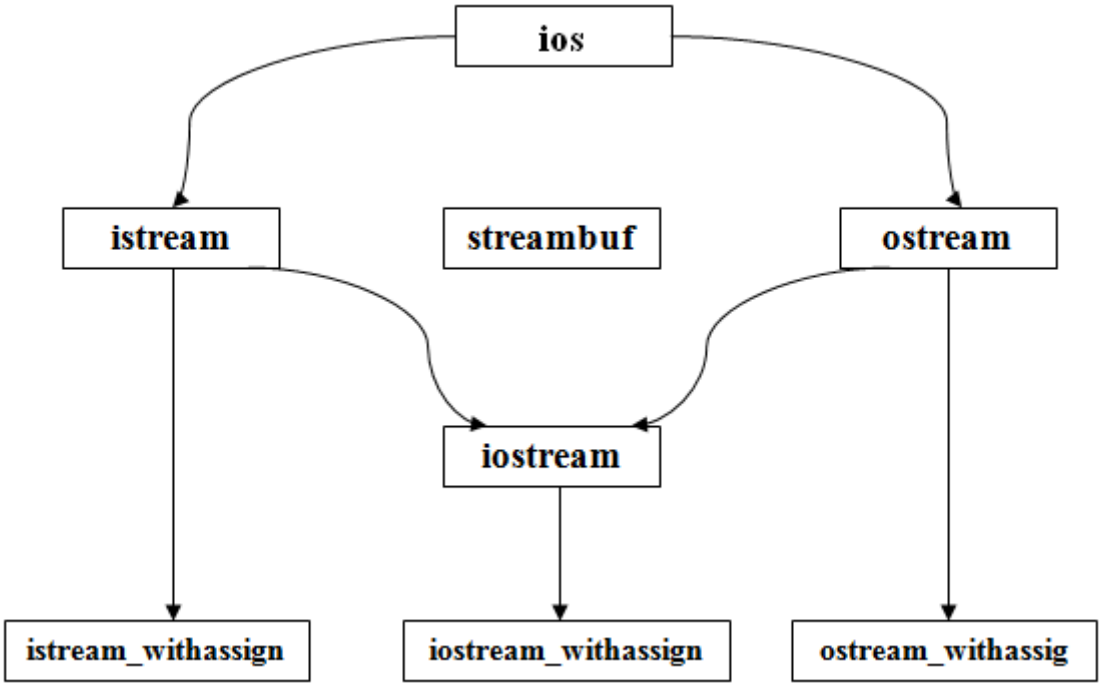
- cout is default connected to monitor
- cout is used to print output on monitor.



cout stream

VF - TECHNIC

6.2 C++ Stream Classes



C++ Stream Classes

- **ios class:**
  - It provides the input & output facilities.
  - It contains a pointer to a buffer object.
- **istream class:**
  - Inherits the properties of ios.
  - Declares input functions such as **get**( ), **getline**( ) and **read**( )
  - Contains overloaded extraction operator >>
- **ostream class:**
  - Inherits the properties of ios.
  - Declares input functions such as put( ) and write( )
  - Contains overloaded extraction operator <<

VPMP POLYTECHNIC

C++ Stream Classes

- **iostream class:**
  - It inherits the properties of ios, istream and ostream through multiple inheritances and thus contains all the input and output functions.

- **streambuf:**
  - Provides an interface to physical devices through buffers.
  - Acts as a base for filebuf class used ios files.

VPMP POLYTECHNIC

6.3 Unformatted I/O Operations

Operators >> and <<

- The >> operator is overloaded in the istream class and the << is overloaded in the ostream class.
- Syntax for reading data for the keyboard:

  cin>> var1>>var2>>var3;

- Syntax for displaying data on the screen is:

  cout<<var1<<var2<<var3;

VPMP POLYTECHNIC

UNFORMATTED I/O OPERATIONS

put( ) and get( ) function

- The classes istream ans ostream define two member functions get( ) and put( ) respectively to handle the single character input/output operations.
- There are two types of get( ) functions.

1) get(char *) version assigns the input character to its argument
2) get(void) version returns the input character.

Example:

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

- The function put( ), a member of ostream class, can be used to output a line of text, character by character.
- The put( ) function is used with the help of the cout object.

Example:

```
cout.put(c);
```

VPMP POLYTECHNIC

UNFORMATTED I/O OPERATIONS

getline( ) functions

- The getline( ) function reads a whole line of text that ends with a newline character.
- The getline( ) function can be invoked by using the object cin.

Syntax**:**

```
cin.getline();
```

Example:

```
char name[20];
cin.getline(name,20);
```

VPMP POLYTECHNIC

FORMATTED I/O OPERATIONS

The ios format functions are:

- **width( )**
- **precision( )**
- **fill( )**

VPMP POLYTECHNIC

width( )

- The width( ) function is used to define the width of the field for the output of an item.
- It is invoked with the help of cout object.

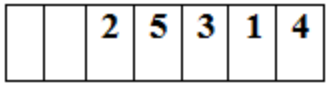
Syntax:

```
cout.width(w);
```

Example:

```
cout.width(5);
cout<<253<<14;
```

Output :



VPMP POLYTECHNIC

PROGRAM

```
#include<iostream.h>
void main( )
{
                    cout.width(5);
                    cout<<253;
                    cout.width(5);
                    cout<<14;
                     cout.width(5);
                    cout<<3;
}
```

Output:

```
  253
   14
    3
```

VPMP POLYTECHNIC

precision( )

- We can specify the numbers of digits to be displayed after the decimal point while printing the floating-point numbers.
- It can be done using the precision( ) member function.

Syntax:

```
cout.precision(d);
```

Example:

```
cout.precision(2);
cout<< 15.2534<<endl;
cout<< 5.4003<<endl;
```

Output:

```
15.25
5.4
```

VPMP POLYTECHNIC

PROGRAM

```
#include<iostream.h>
void main( )
{
        cout.precision(2);
       cout<< 11.4321<<endl;
       cout<< 5.4003<<endl;
       cout<<12.333333<<endl
    }
```

Output:

11.43

5.4

12.33

VPMP POLYTECHNIC

fill( )

- When we print the values using much larger field with than required by the values.
- By default, the unused positions of the field are filled with whitespaces.

Syntax:

```
cout.fill(ch);
```

Example:

```
cout.fill('*');
cout.width(5);
cout<<25;
```

Output:

*	*	*	2	5

VPMP POLYTECHNIC

CONT…

```
#include<iostream.h>
void main( )
{
                cout.fill('*');
                cout.width(10);
                cout<<54321;
                cout.fill('*');
                cout.width(10);
                cout<<21;
}
```

Output:

*****54321

********21

VPMP POLYTECHNIC

6.4 Formatting with Manipulators

- The header file **iomanip** provides a set of functions called manipulators which can be used to manipulate the output formats.

Manipulators	Meaning	Equivalent
setw(int w) setprecision(int d)	Set the field width to w. Set the floating point precision to d.	width( ) precision( )
setfill(int c)	Set the fill character to c	fill( )
setiosflags(long f)	Set the format flag f.	setf( )
resetiosflags(long f)	Clear the flag specified by f	unsetf( )
endl	Insert new line	"\n"

SETW( )

- This manipulator sets the minimum field width on output.

- Syntax:

  setw(x);

Example:

```
#include <iostream>
void main( )
{
   float basic, ta,da,gs;
   basic=10000;
```

VPMP POLYTECHNIC

CONT…

```
ta=800;
da=5000;
gs=basic+ta+da;
cout<<setw(10)<<"Basic"<<setw(10)<<basic<<endl
   <<setw(10)<<"TA"<<setw(10)<<ta<<endl
   <<setw(10)<<"DA"<<setw(10)<<da<<endl
   <<setw(10)<<"GS"<<setw(10)<<gs<<endl;
}
```

Output:

```
   Basic        10000
      TA          800
      DA         5000
      GS        15800
```

VPMP POLYTECHNIC

SETFILL( )

- This is used after setw manipulator.
- If a value does not entirely fill a field, then the character specified in the setfill argument of the manipulator is used for filling the fields.
- Syntax:

```
setfill(char);
```

VPMP POLYTECHNIC

EXAMPLE

```
   #include <iostream>
void main( )
 {
        cout<< setfill('0');
        cout<< setw(10) <<11<<"\n";
        cout<< setw(10) <<2222<<"\n";
        cout<< setw(10) <<4<<"\n";
}
```

Output:

```
   0000000011
    0000002222
    0000000004
```

VPMP POLYTECHNIC

SETPRECISION( )

- Sets the *decimal precision* to be used to format floating-point values on output operations.
- Behaves as if member precision were called with n as argument on the stream on which it is inserted/extracted as a manipulator
- This manipulator is declared in header <iomanio>.

VPMP POLYTECHNIC

EXAMPLE

```
#include <iostream>
#include <iomanip>
void main ( )
{
    double f =3.14159; c
    cout << std::setprecision(4) << f<<'\n';
    cout << std::setprecision(5) << f << '\n';
    cout << std::setprecision(5) << f << '\n';
    cout << std::setprecision(9) << f << '\n';
}
```

Output:

```
3.1416
3.14159
3.14159
3.141590000
```

VPMP POLYTECHNIC

Thanks…