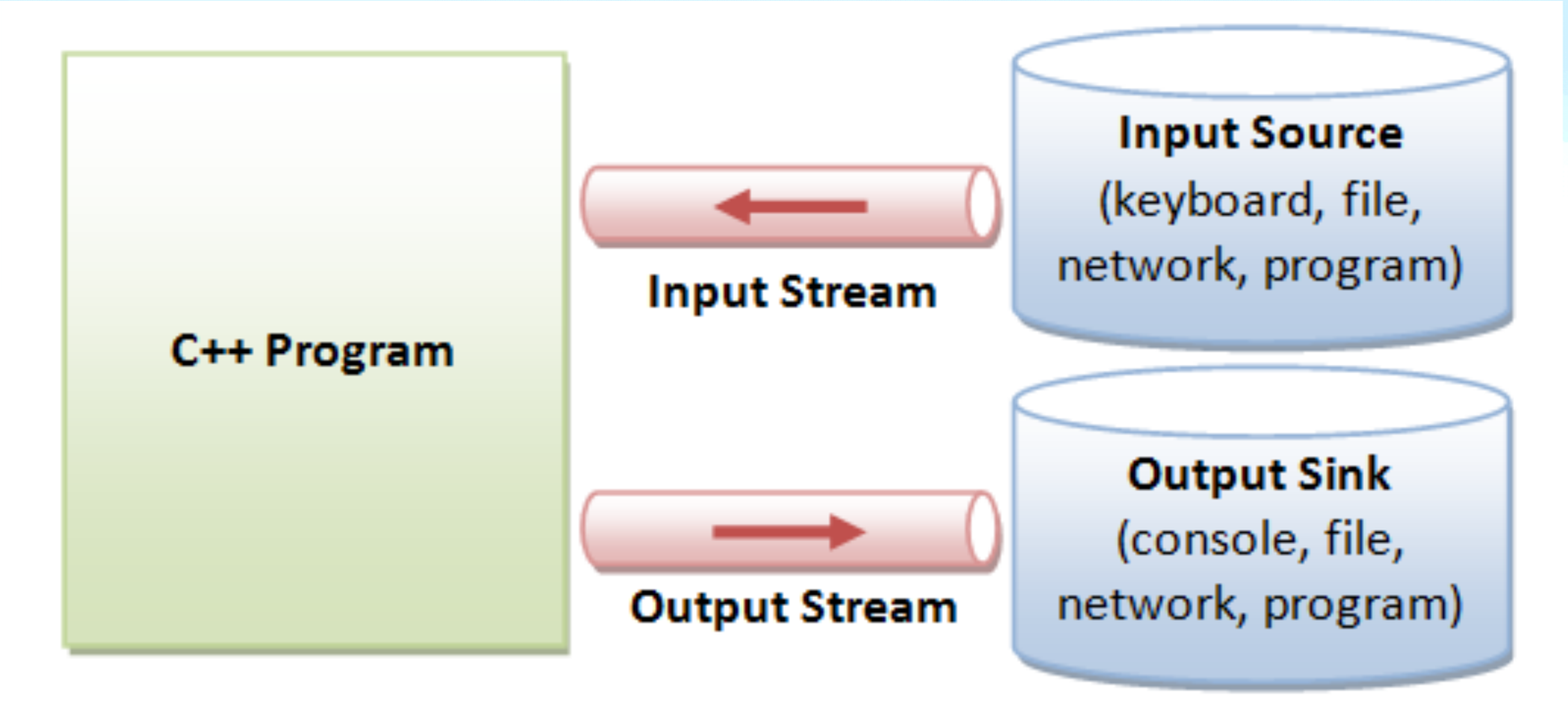
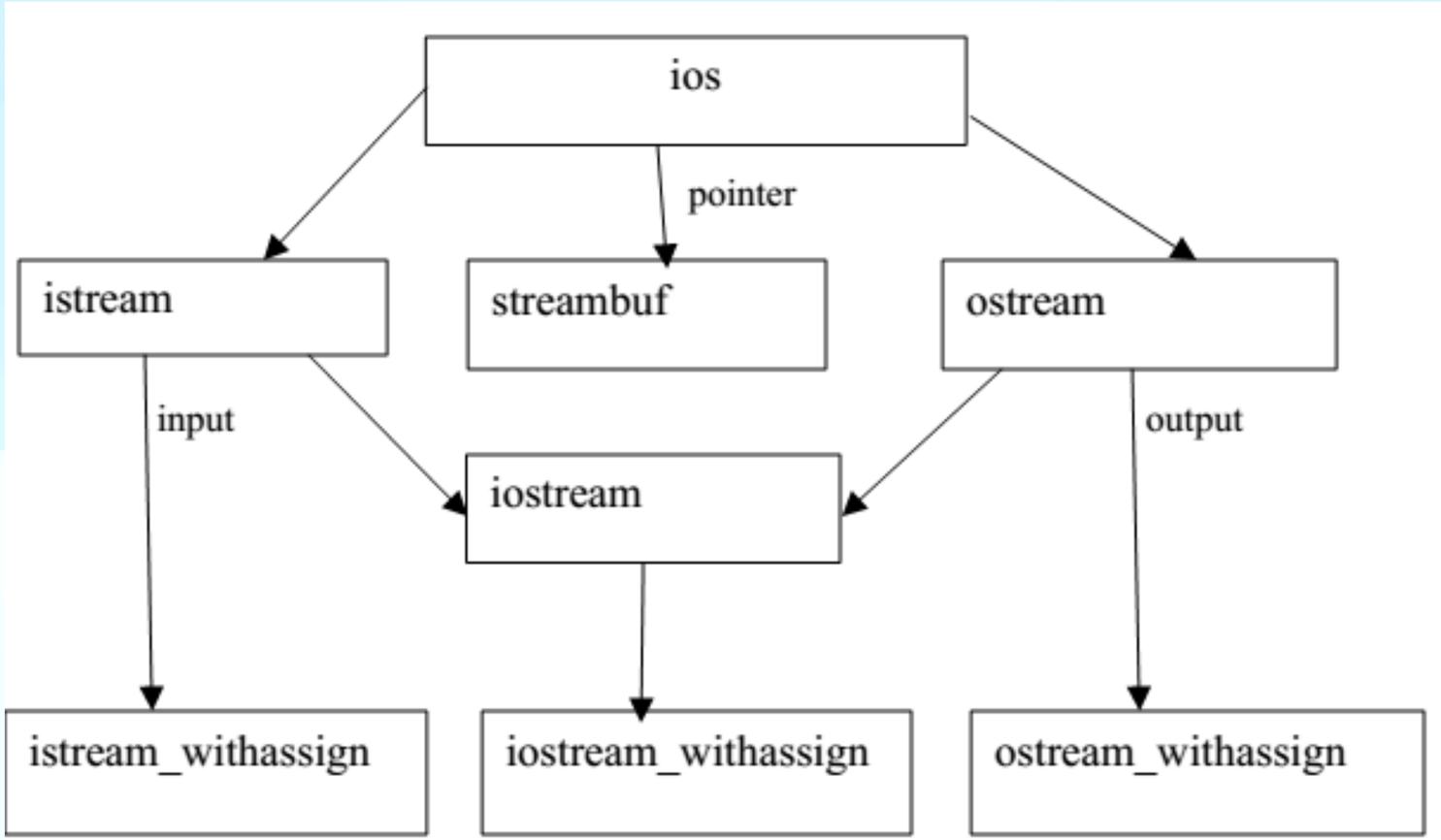
Managing Console I/O Operations

Zankhana Dabhi
Information Technology
DDU, Nadiad

Data Streams



C++ Stream Classes



Stream classes for console I/O operations

Unformatted I/O Operations

- Overloaded Operators >> and <<

cin>>var;

cout<<var;

Unformatted I/O Operations

- put() and get() functions(read with whitespace and newline character)

```
cin.get(c);   OR      c=cin.get();

cout.put(c);

cout.put(68);
```

Unformatted I/O Operations

- getline() and write() functions:(line oriented functions)

```
char name[20];
cin.getline(name,20);
cout.write(name, 20);       //displays string upto length 20
cout.write(name1,20).write(name2,20);
```

Formatted Console I/O Operations

- Features to be used for formatting output:
- iOS class functions and flags
- Manipulators
- User defined output functions

iOS format functions

- width(): To specify the required field size for displaying an output value.

```
cout.width(5);
Cout<<543<<12;      // Output:      54312


cout.width(5);
Cout<<543;
cout.width(5);
Cout<<12;              // output:      543    12
```

iOS format functions

- precision(): To specify the number of digits to be displayed after decimal point of a float value.

```
cout.precision(3);     //  output
cout<<sqrt(2);         //  1.141
cout<<3.14159          //  3.142
cout<<2.50032          //  2.5


cout.precision(3);     //  output
cout<<sqrt(2);         //  1.141
cout.precision(5);
cout<<3.14159          //  3.1416
```

iOS format functions

- fill(): To specify a character that is used to fill the unused portion of a field.

```
Cout.fill('*');

Cout.width(10);

Cout<<5250;
```

Output:

* * * * * * 5 2 5 0

iOS format functions

- setf(): To specify format flags that can control the form of output display. (Such as left-justification and right-justification)

cout.setf(arg1,arg2)

  - arg1 is one of the formatting flags defined in the class iOS.
  - arg2 is known as bit field, specifies the group to which the formatting flag belongs.

iOS format functions

- setf(): To specify format flags that can control the form of output display. (Such as left-justification and right-justification)

Format	Flag	Bit Field
Left justification	ios::left	ios::adjustfield
Right justification	ios::right	ios::adjustfield
Padding after sign and base	ios::internal	ios::adjustfield
Scientific notation	ios::scientific	ios::floatfield
Fixed point notation	ios::fixed	ios::floatfield
Decimal base	ios::dec	ios::basefield
Octal base	ios::oct	ios::basefield
Hexadecimal base	ios::hex	ios::basefield

iOS format functions

- setf(): To specify format flags that can control the form of output display. (Such as left-justification and right-justification)

  Example1:

  Cout.fill(‘*’);

  Cout.setf(iOS::left, iOS::adjustfield);

  Cout.width(15);

  Cout<<“TABLE 1”;

  Output:

  T A B L E 1 * * * * * * * *

iOS format functions

- setf(): To specify format flags that can control the form of output display. (Such as left-justification and right-justification)

  Example2:

  Cout.fill(‘*’);

  Cout.precision(3);

  Cout.setf(iOS::internal, iOS::adjustfield);

  Cout.setf(iOS::scientific, iOS::floatfield);

  Cout.width(15);

  Cout<<-12.34567;

  Output:

  - * * * * * 1 . 2 3 5 e + 0 1

iOS format functions

- setf(): To specify format flags that can control the form of output display. (Such as left-justification and right-justification)
- Flags that do not have bit fields

Flag	Purpose
Ios::showbase	Uses base indicator on output.
Ios::showpos	Displays + preceding positive number.
Ios::showpoint	Shows trailing decimal point and zeros.
Ios::uppercase	Uses capital case for output.
Ios::skipws	Skips white space on input.
Ios::unitbuf	Flushes all streams after insertion.
Ios::stdio	Adjusts the stream with C standard input and output.
ios::boolapha	Converts boolean values to text.

iOS format functions

- setf(): To specify format flags that can control the form of output display. (Such as left-justification and right-justification)

  iOS::showpoint() : show trailing zeros

  ```
  cout.precision(2);

  Cout<<10.75;    //  10.75

  Cout<<25.00;    //   25

  Cout<<15.50;    //   15.5
  ```

iOS format functions

- setf(): To specify format flags that can control the form of output display. (Such as left-justification and right-justification)

  iOS::showpoint() : show trailing zeros

  Cout.setf(iOS::showpoint);

  Cout.setf(iOS::showpos);

  cout.precision(3);

  cout.setf(iOS::fixed,ios::floatfield);

  cout.setf(iOS::internal,ios::adjustfield);

  Cout.width(10);

  Cout<<275.5;

  Output: + 2 7 5 . 5 0 0

Managing output with manipulators

The most commonly used manipulators are shown in table.

Manipulator	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()
Endif	Insert new line and flush stream	“\n”