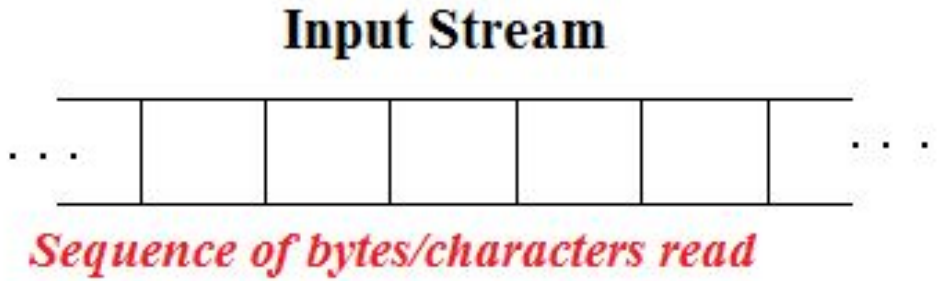
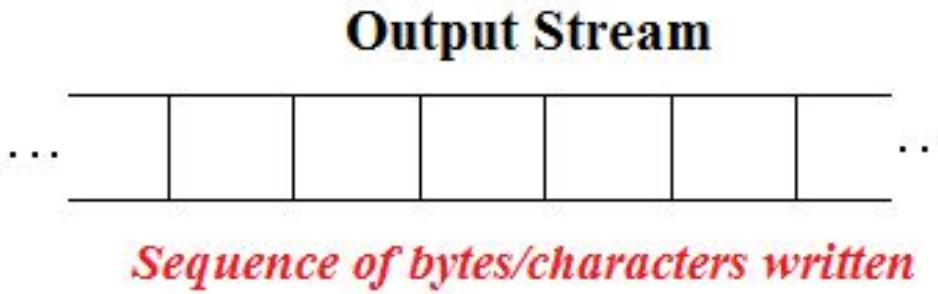
Object-oriented Programming

Week 10 | Lecture 2

Stream

Stream (for us…)





istream/ostream Member Functions

- Both the ***istream*** and ***ostream*** classes provide member functions for input/output of data

- These member functions can be called using the ***iostream*** objects such as ***cin*** & ***cout***

Reading a character from file

get()

The **get()** member function of ifstream class reads a single character from the file input stream (including white-space and other characters like EOF) and returns it

Reading a character from file

get()

Example:

```
ifstream in("xyz.txt");
char c = in.get();
cout << c;
```

Reading a character from file

get(char)

The **get(char)** member function of *ifstream* class reads a single character from the file input stream (including white-space and other characters like EOF) in the character variable specified in the argument

Reading a character from file

get(char)

Example:

```
ifstream in("xyz.txt");
char c;
in.get(c);
cout << c;
```

Reading a line from file

getline(ifstream, string)

The function **getline(ifstream, string)** reads a single line from the file input stream specified as first argument and saves it in the string specified as the second argument

Reading a line from file

getline(ifstream, string)

Example:

```
ifstream in("xyz.txt");
string line;
getline(in, line);
cout << line;
```

Reading bulk text from a file

read(char, int)*

The function **read(char*, int)** reads n characters (including whitespaces and eof) into the buffer specified as first argument. The value of n is specified as second argument

Reading bulk text from a file

read(char, int)*

Example:

```
ifstream in("xyz.txt");
char* text;
int n = 20;
in.read(text, n);
```

Writing a character to file

put(char)

The **put(char)** member function of ofstream class writes a single character, taken as argument, to the file specified by file output stream object

Writing a character to file

put(char)

Example:

```
ofstream out("xyz.txt");
out.put('A');
char c = 'B';
out.put(c);
```

Writing bulk text to a file

write(char*, int)

The function **write(char*, int)** writes n characters from the char* buffer specified as first argument to the file. The value of n is specified as second argument

Writing bulk text to a file

write(char*, int)

Example:

```
ofstream out("xyz.txt");
out.write("This is some text", 8);
char* c = "This is some other text";
out.write(c, 10);
```

istream Member Function

eof()

- It returns **1** (TRUE) when there are no more data to be read from an input stream, and **0** (FALSE) otherwise

```
 int main()
{
int character;
cout << “Before input, cin.eof() is ”<< cin.eof();
cout << “Enter input followed by eof” <<  endl;

while((character = cin.get())  != EOF)
 cout.put( character );

cout << “EOF in this system is: ” << character;
cout << “After input, cin.eof() is: ” << cin.eof();
}
```

Output

```
Before input, cin.eof() is 0
Enter a sentence followed by end-of-file:
Testing the get and put member functions
Testing the get and put member functions
^Z

EOF in this system is: -1
After input of EOF, cin.eof() is 1
```

istream Member Function

ignore()

- The **ignore()** function of istream reads and discards a designated number of characters (the default is one) or terminates upon encountering a designated delimiter

istream Member Function

putback()

- The **putback()** function places the previous character obtained by a get from an input stream back into that stream

istream Member Function

peek()

- The **peek()** function returns the next character from an input stream but does not remove the character from the stream

Notes

- The default **end-of-file (EOF)** character sequence in Windows is ***Ctrl+Z***

- The default delimiter in most systems is **'\n'** or newline character