Object-Oriented Programming (OOP)

Week – 12

April 27-30, 2020

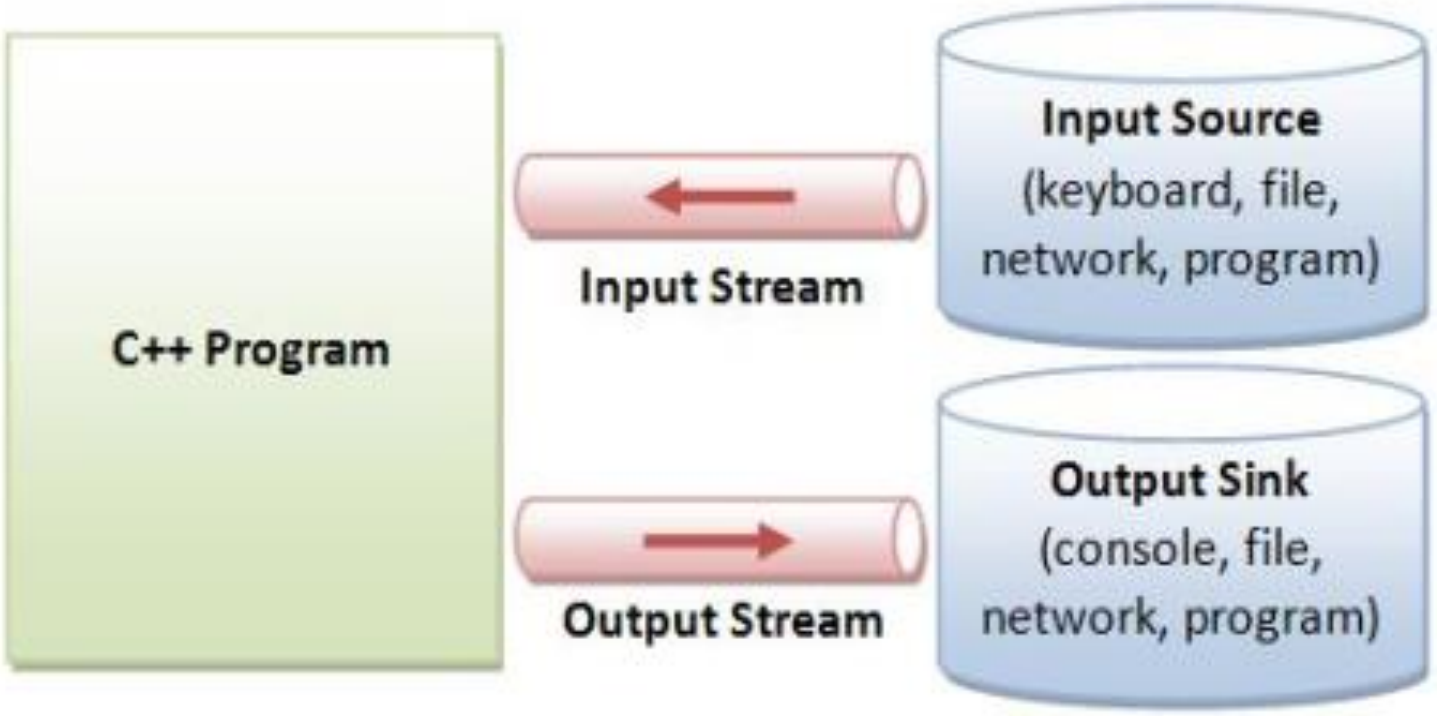
Instructor: **Basit Ali**

Email: basit.jasani@nu.edu.pk

Object-Oriented Programming (OOP)

Streams

- **Stream**
  - A transfer of information in the form of a sequence of bytes
- **I/O Operations**
  - Input: A stream that flows from an input device (i.e.: keyboard, disk drive, network connection) to main memory
  - Output: A stream that flows from main memory to an output device (i.e.: screen, printer, disk drive, network connection)


C++ Program
Input Stream
Input Source
(keyboard, file,
network, program)
Output Stream
Output Sink
(console, file,
network, program)

Iostream Library Header Files

`iostream` library

- **`<iostream.h>`:** Contains **`cin & cout`** objects
- **`<fstream.h>`:** Contains information important to user-controlled file processing operations

file (fstream)

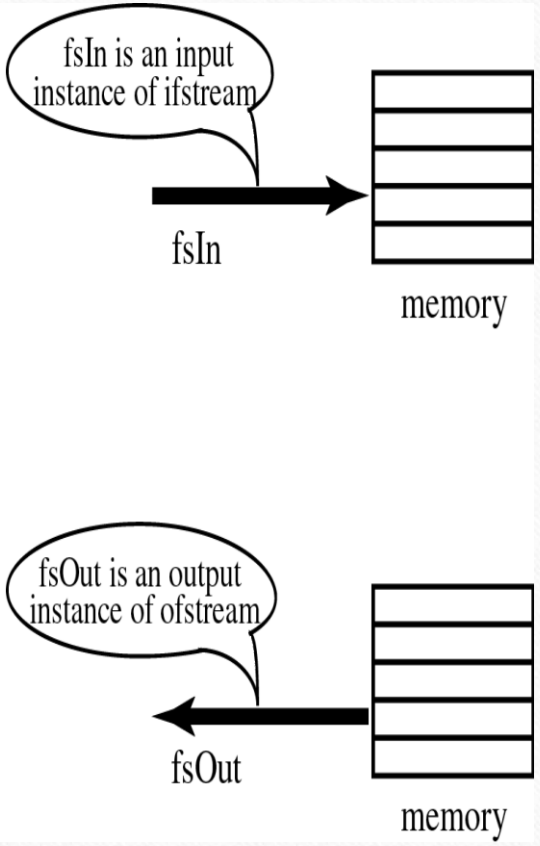
- **ifstream** - defines new input stream (normally associated with a file).
- **ofstream** - defines new output stream (normally associated with a file).

General File I/O Steps

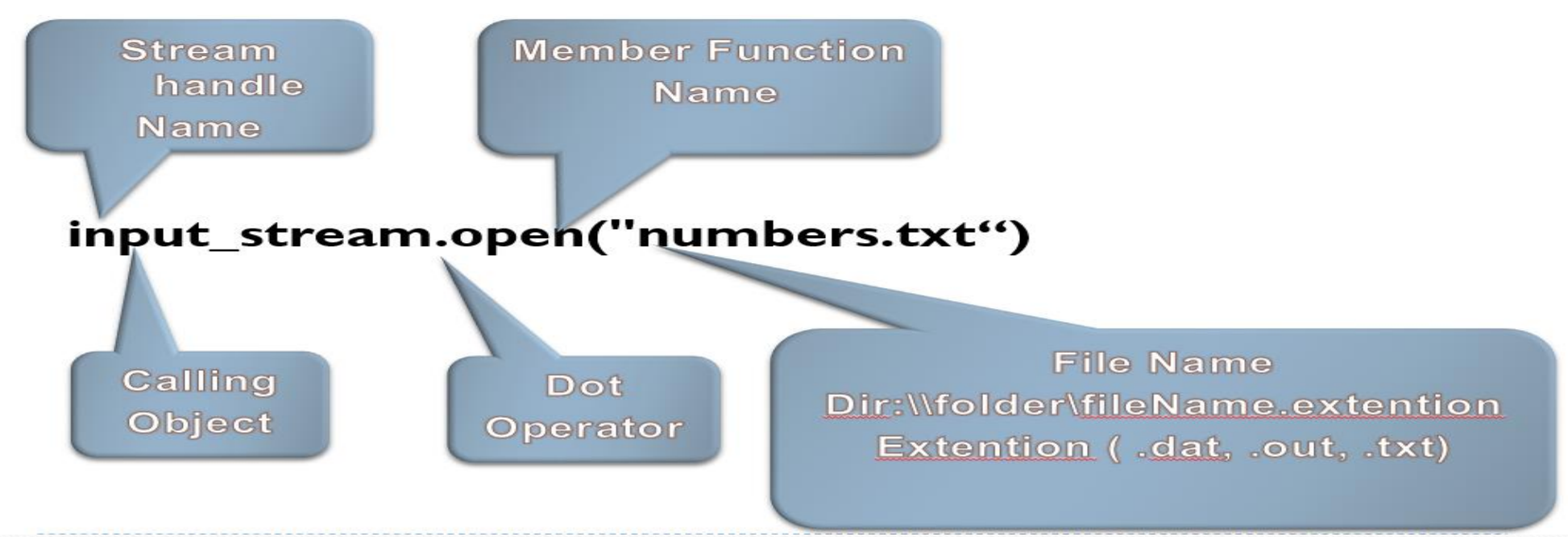
1. Include the header file fstream in the program.
2. Declare file stream variables.
3. Open the file
4. Use the file stream variables with >>, <<, or other input/output functions.
5. Close the file.

```
2   #include<iostream>
3   #include <string>
4   #include <fstream>
5   using namespace std;
6
7   int main ()
8   {
9   /* Declare file stream variables such as
10   the following  */
11
12  ifstream  fsIn;//input
13  ofstream fsOut; // output
14  fstream both; //input & output
15
16  //Open the files
17  fsIn.open("prog1.txt"); //open the input file
18  fsOut.open("prog2.txt"); //open the output file
19
20  //Code for data manipulation
21  //Close files
22  fsIn.close();
23  fsOut.close();
24  return 0;
25  }
```

```
#include <fstream.h>

int main (void)
{
// Local Declarations
   ifstream     fsIn;

   ofstream     fsOut;
   •
   •
   •

} // main
```



Object and Member Functions



Open()

- Opening a file associates a file stream variable declared in the program with a physical file at the source, such as a disk.
- In the case of an input file:
  - the file must exist before the open statement executes.
  - If the file does not exist, the open statement fails and the input stream enters the fail state
- An output file does not have to exist before it is opened;
  - if the output file does not exist, the computer prepares an empty file for output.
  - If the designated output file already exists, by default, the old contents are erased when the file is opened.

Validate the file before trying to access

Method 1:

By checking the stream variable;

```
If ( ! Mystream)
{
Cout << "Cannot open file.\n ";
}
```

Method 2:

By using bool is_open() function.

```
If ( ! Mystream.is_open()) {
Cout << "File is not open.\n ";
}
```

Input File-Related Functions

- **fsin.get(char character)**

extracts next character from the input stream fsin and places it in the character variable character.

- **fsin.eof()**

tests for the end-of-file condition.

File I/O Example: Reading

- Read char by char

```
#include <iostream>
#include <fstream>
int main()
{//Declare and open a text file
ifstream openFile(“data.txt");
 char ch;
 //do until the end of file
while( ! OpenFile.eof() )
{
OpenFile.get(ch); // get one character
cout << ch;   // display the character
}
OpenFile.close(); // close the file
    return 0;
}
```

- Read a line

```
#include <iostream>
#include <fstream>
#include <string>
int main()
{//Declare and open a text file
ifstream openFile("data.txt");
string line;
while(!openFile.eof())
{//fetch line from data.txt and put it in a string
getline(openFile, line);
cout << line;
}
openFile.close(); // close the file
    return 0; }
```

Output File-Related Functions

- **ofstream fsOut;**
- **fsOut.open(const char[] fname)**

  connects stream fsOut to the external file fname.
- **fsOut.put(char character)**

  inserts character character to the output stream fsOut.
- **fsOut.eof()**

  tests for the end-of-file condition.

File I/O Example: Writing

- First Method (use the constructor)

```
#include <fstream>
using namespace std;
int main()
{/* declare and automatically open the file*/
ofstream outFile("fout.txt");

//behave just like cout, put the word into the file
outFile << "Hello World!";
outFile.close();
return 0;
}
```

- Second Method (use Open function)

```
#include <fstream>
using namespace std;
int main()
{// declare output file variable
ofstream outFile;
// open an exist file fout.txt
    outFile.open("fout.txt”);

//behave just like cout, put the word into  the file
outFile << "Hello World!";

outFile.close();

return 0;
}
```

File Open Mode

Name	Description
ios::in	Open file to read
ios::out	Open file to write
ios::app	All the data you write, is put at the end of the file. It calls ios::out
ios::ate	All the data you write, is put at the end of the file. It does not call ios::out
ios::trunc	Deletes all previous content in the file. (empties the file)
ios::nocreate	If the file does not exists, opening it with the open() function gets impossible.
ios::noreplace	If the file exists, trying to open it with the open() function, returns an error.
ios::binary	Opens the file in binary mode.

File Open Mode

```
#include <fstream>
int main(void)
{
ofstream outFile("file1.txt", ios::out);
outFile << "That's new!\n";
outFile.close();
        Return 0;
}
```

If you want to set more than one open mode, just use the **OR** operator- **|**. This way:

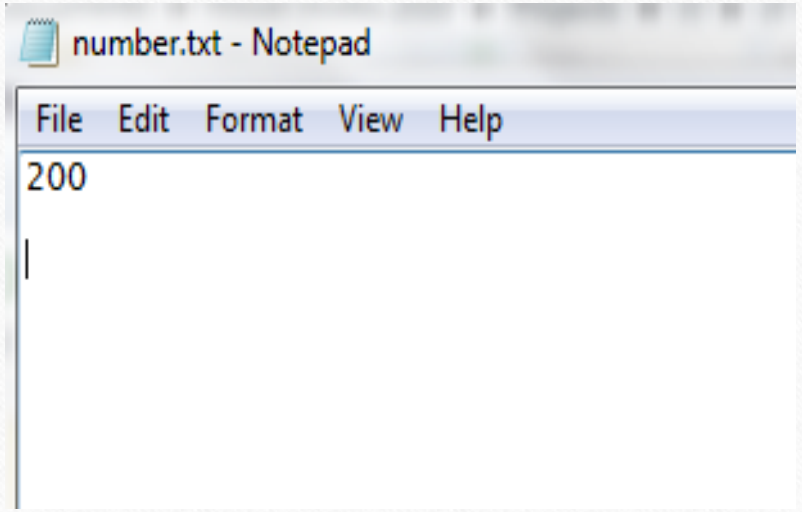
ios::ate | ios::binary

File format

- In c++ files we (read from/ write to) them as a stream of characters
- What if I want to write or read numbers ?

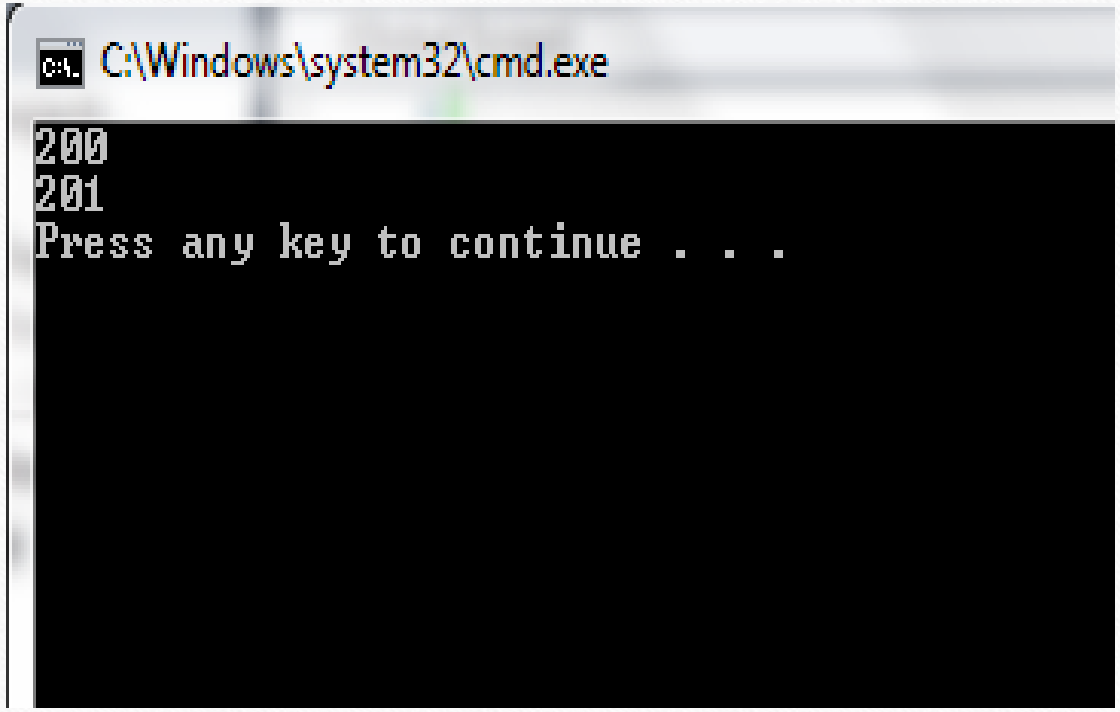
Example writing to file

```
#include <iostream>
#include <fstream>
using namespace std;
void  main()
{
ofstream outFile;
// open an exist file fout.txt
    outFile.open("number.txt",ios::app);

if (!outFile.is_open())
{ cout << " problem with opening the file ";}
else
{outFile <<200 <<endl ;
cout << "done writing" <<endl;}

outFile.close();

}
```



Example Reading from file

```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
using namespace std;
void main()
{//Declare and open a text file
ifstream INFile("number.txt");
string line;
int total=0;
while(! INFile.eof())
{
getline(INFile, line);
//converting line string to int
stringstream(line) >> total;
cout << line  <<endl;
cout <<total +1<<endl;}
INFile.close(); // close the file
 }
```



Writing OBJECTS to file

```
15  int main()
16  {
17      A a;
18      a.x = 10;
19      a.y = 20;
20
21      ofstream obj;
22
23      obj.open("C:/Users/basit.jasani/Desktop/abc.bin",ios::out | ios::binary | ios::trunc);
24      obj.write((char*)&a,sizeof(a));
25      obj.close();
26
27      ifstream obj2;
28
29      A b;
30
31      obj2.open("C:/Users/basit.jasani/Desktop/abc.bin",ios::in);
32      obj2.read((char*)&b,sizeof(b));
33
34      cout << b.x << " " << b.y << endl; //}
35      obj2.close();
36  }
```

```
8   class A
9   {
10      public:
11      int x;
12      int y;
13  };
```