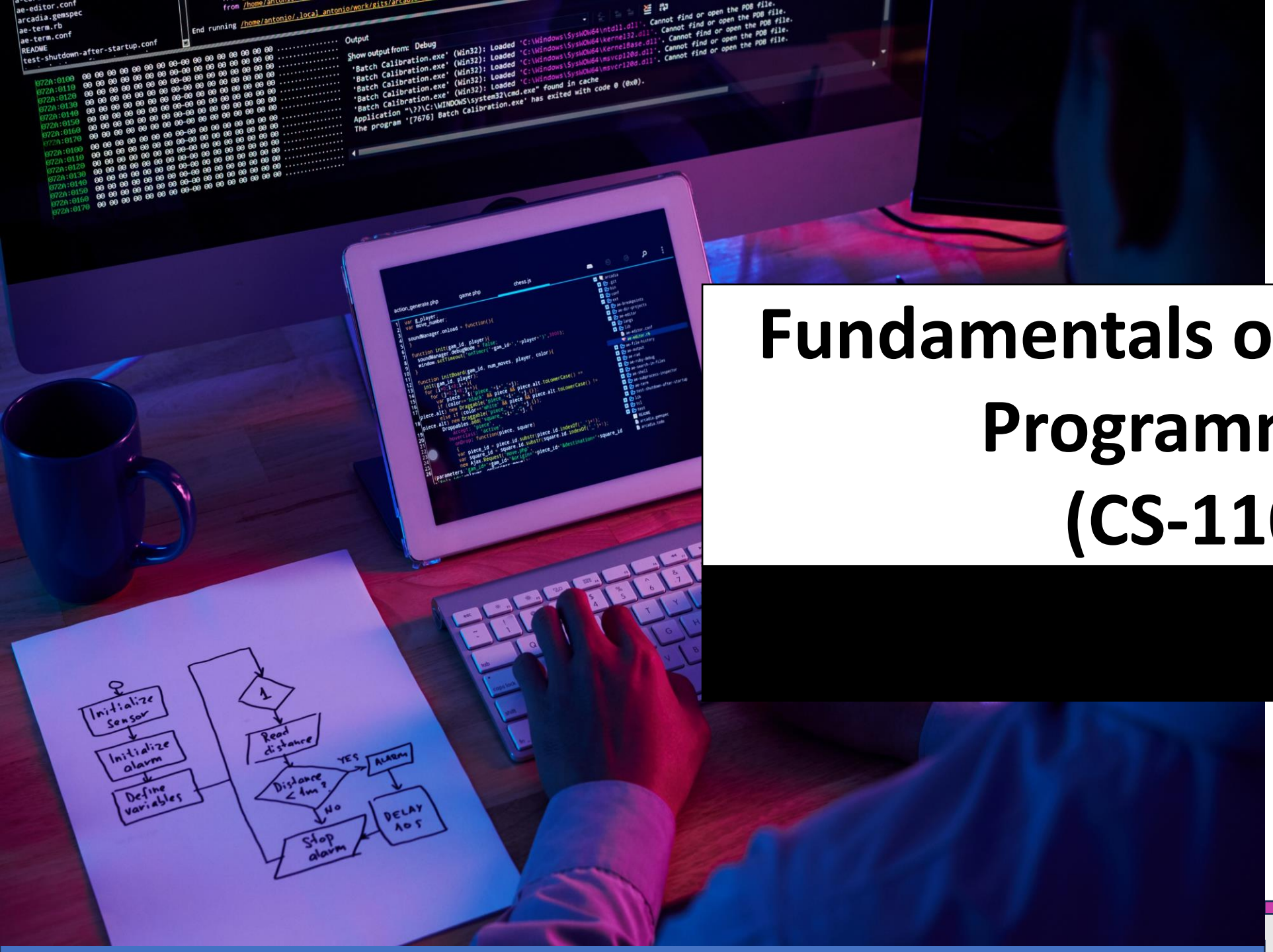


# Fundamentals of Computer Programming (CS-110)

Course Instructors:
Dr. Momina Moetesum
Ms. Shakeela Bibi

# File Handling

- File handling is used to store data permanently in a computer.
- Using file handling we can store our data in secondary memory (Hard disk).
- How to achieve the File Handling
- For achieving file handling we need to follow the following steps:-

STEP 1-Naming a file
STEP 2-Opening a file
STEP 3-Writing data into the file
STEP 4-Reading data from the file
STEP 5-Closing a file.

# File Handling in C++

- The **<fstream>** library allows us to work with files.
- To use the **<fstream>** library, include both the standard **<iostream>** AND the **<fstream>** header file

## Example

```
#include <iostream>
#include <fstream>
```

# Fstream Library

There are three classes included in the fstream library, which are used to create, write or read files:

| Class | Description |
|---|---|
| `ofstream` | Creates and writes to files |
| `ifstream` | Reads from files |
| `fstream` | A combination of ofstream and ifstream: creates, reads, and writes to files |

# fstream

This class provides support for simultaneous input and output operations.

Inherits all the functions from istream and ostream classes through iostream.

# ifstream

This class provides input operations.

It contains open() function with default input mode.

Inherits the functions get(), getline(), read(), seekg() and tellg() functions from the istream.

# ofstream

This class provides output operations.

It contains open() function with default output mode.

Inherits the functions put(), write(), seekp() and tellp() functions from the ostream.

# Opening a File

The first step to open the particular file for read or write operation. We can open file by

1. passing file name in constructor at the time of object creation
2. using the open method

For e.g.

- Open File by using constructor
  - ifstream (const char* filename, ios_base::openmode mode = ios_base::in);
  - ifstream fin(filename, openmode) by default openmode = ios::in
  - ifstream fin(“filename”);

Open File by using open method

- Calling of default constructor
- ifstream fin;
- fin.open(filename, openmode)
- fin.open(“filename”);

# Create and Write To a File

- To create a file, use either the **ofstream** or **fstream** class, and specify the name of the file.
- To write to the file, use the insertion operator (<<).

```
#include <iostream>
#include <fstream>
using namespace std;

int main() {
  // Create and open a text file
  ofstream MyFile("filename.txt");

  // Write to the file
  MyFile << "Files can be tricky, but it is fun enough!";

  // Close the file
  MyFile.close();
}
```

# Read a File

```
// Create a text string, which is used to output the text file
string myText;

// Read from the text file
ifstream MyReadFile("filename.txt");

// Use a while loop together with the getline() function to read the file line by line

while (getline (MyReadFile, myText)) {
  // Output the text from the file
  cout << myText;
}

// Close the file
MyReadFile.close();
```

- To read from a file, use either the **ifstream** or **fstream** class, and the name of the file.
- We also need a loop together with the getline() function (which belongs to the ifstream class) to read the file line by line, and to print the content of the file:

# Complete Example

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main () {
  // Create a text file
  ofstream MyWriteFile("filename.txt");

  // Write to the file
  MyWriteFile << "Files can be tricky, but it is fun enough!";

  // Close the file
  MyWriteFile.close();

  // Create a text string, which is used to output the text file
  string myText;

  // Read from the text file
  ifstream MyReadFile("filename.txt");

  // Use a while loop together with the getline() function to read the file line by line
  while (getline (MyReadFile, myText)) {
    // Output the text from the file
    cout << myText;
  }
  // Close the file
  MyReadFile.close();
  return 0;
}
```

# Modes

| Member Constant | Stands For | Access |
|---|---|---|
| in * | input | File open for reading: the internal stream buffer supports input operations. |
| out | output | File open for writing: the internal stream buffer supports output operations. |
| binary | binary | Operations are performed in binary mode rather than text. |
| ate | at end | The output position starts at the end of the file. |
| app | append | All output operations happen at the end of the file, appending to its existing contents. |
| trunc | truncate | Any contents that existed in the file before it is open are discarded. |

| | |
|---|---|
| ifstream | ios::in |
| ofstream | ios::out |
| fstream | ios::in \| ios::out |

# Ifstream Functions

ifstream – Provides functions for input operations. It inherits all the functions from istream class.

- get ( )
- getline ( )
- read ( )
- seekg ( )
- tellg ( )
- open ( ) with default input mode

# Pointer offset calls

| Seek calls | Action |
|---|---|
| fout.seekg(0,ios::beg) | Go to start |
| Fout.seekg(0,ios::cur) | Stay at the current position |
| Fout.seekg(0,ios::end) | Go to end of the file |
| fout,.seekg(m,ios::beg) | Moves to (m+1) th byte in the file |
| Fout.seekg(m,ios::cur) | Go forward by m bytes from current posiotion |
| Fout.seekg(-m,ios::cur) | Go backward by m bytes from the current position |
| Fout.seekg(-m,ios::end) | Go backward by m bytes from end of the file |

# Example of read() and seekg() with ios::beg

```cpp
#include <iostream>
#include <fstream>
using namespace std;

int main (int argc, char** argv){
   fstream File("d.txt", ios::in | ios::out | ios::trunc);
   File << "Hello World";
   File.seekg(0, ios::beg);
   char F[6];
   File.read(F, 5);
   F[5] = 0;
   cout <<F<< endl;
   File.close();
}
```

Output

```
/tmp/WjzFKP8Hl1.o
Hello
```

# Example of read() and seekg() with ios::end

```
// Online C++ compiler to run C++ program online
#include <iostream>
#include <fstream>
using namespace std;

int main (int argc, char** argv){
   fstream File("d.txt", ios::in | ios::out | ios::trunc);
   File << "Hello World";
   File.seekg(-6, ios::end);
   char F[7];
   File.read(F, 6);
   F[6] = 0;
   cout <<F<< endl;
   File.close();
}
```

Output

```
/tmp/WjzFKP8Hl1.o
World
```

Example of tellg()

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main ()
{
  fstream obj;
  obj.open ("test.txt", ios::out |ios::trunc);
  obj<<"Hello World";
  obj.close();
  obj.open("test.txt", ios::in);
  char ch;
  int pos;
  while(!obj.eof())
  {
    obj>>ch;
    pos = obj.tellg();
    cout<<pos<<"."<<ch<<"\n";
  }
  obj.close();
}
```

Output

```
/tmp/WjzFKP8H11.o
1.H
2.e
3.l
4.l
5.o
7.W
8.o
9.r
10.l
11.d
-1.d
```

# ofstream Functions

ofstream – Provides functions for output operations. It inherits all the functions from ostream class.

- put ( )
- write ( )
- seekp ( )
- tellp ( )
- open ( ) with default output mode

fstream – Provides support for input and output both operations. It inherits all the functions from istream and ostream classes through iostream.

# Example of Seekp()

```cpp
#include <fstream>
#include <iostream>
using namespace std;
int main () {
   fstream file("cppsecret.txt", ios::in | ios::out |ios::trunc);  //create file
   string str;
   file.write ("This is a cpp site",18);                           //write 18 characters in file
   int pos = file.tellp();                                         //tell position of put pointer
   cout<<"Current location of pointer="<<pos<<endl;
   //it will be 18 as next character will be written on 19th position
   file.seekg(0,ios::beg);          //take the get pointer to start of file to read from beginning
   while(getline(file,str))         //read from beginning
   {
       cout<<str;
   }
   cout<<endl;
   file.close();                    //close file
   file.open("cppsecret.txt");      //open file
   file.seekp(pos-7);               //move put pointer to 7 characters back from the end
   cout<<"Current location of pointer="<<file.tellp()<<endl;   //display position of put pointer
   file.write (" win",4);           //start writing 4 characters from the put pointer location
   file.seekg(0,ios::beg);          //take the get pointer to start of file to read from beginning
   while(getline(file,str))         //read from beginning
   {
       cout<<str;
   }
   file.close();
   return 0;
```

**Output**

```
/tmp/WjzFKP8Hl1.o
Current location of pointer=18
This is a cpp site
Current location of pointer=11
This is a c winite
```

# Example of tellp()

- In C++ file handling, the tellp() function is used with output streams, and returns the current put position of the pointer in the stream.
- It returns an integer data type, representing the current position of the stream pointer.

```
#include <iostream>
#include <iostream>
#include <fstream>
using namespace std;
int main() {
   fstream newfile;
   newfile.open("test.txt", ios::out); //open a file to perform write operation using file object
   newfile << "Hello World"; //inserting data
   cout << "The present position of the pointer in the file: "
   << newfile.tellp() << endl; //position of the pointer in the file object
   newfile.close(); //close file object.
}
```

Output

```
/tmp/WjzFKP8H11.o
The present position of the pointer in the file: 11
```

# Checking for file open

- If the file does not exist when reading a file, your program will run and produce incorrect results. Can your program check whether a file exists?
- Yes. You can invoke the fail() function immediately after invoking the open function. If fail() returns true, it indicates that the file does not exist.

```
{ input.open("scores.txt");
  if (input.fail()){
     cout << "File does not exist" << endl;
     cout << "Exit program" << endl;
}
```

# Handout

Please Go through the handout provided

# Acknowledgment

- Content of these slides are taken from:
  - https://www.geeksforgeeks.org/
  - https://www.tutorialspoint.com/
  - https://www.programiz.com/
  - https://www.w3schools.com/