Tishk International University
Science Faculty
IT Department

Programming II - IT-118

File stream

1st Grade - Fall Semester

Lecture #5

Instructor: Hemin Ibrahim
Email: hemin.ibrahim@tiu.edu.iq

Overview

- ✓ Files
- ✓ Streams
- ✓ Streams Usage
- ✓ Declaring Streams
- ✓ Open the file for writing
- ✓ Open the file for reading
- ✓ fstream for reading and writing
- ✓ Input string vs numbers
- ✓ Using while loop vs for loop
- ✓ Read from file to vector

Files

- A file is a set of data stored on a computer, often on a disk drive
- Programs can read from, write to files
- Used in many applications:
  - *Word processing*
  - *Databases*
  - *Spreadsheets*
  - *Compilers*

Files (cont..)

- ***Word processing***
  - *Example: Microsoft Word, Google Docs, or Pages*
  - *Contents are stored in a file, and file can be saved on your computer's disk drive or cloud.*
  - *It can be open, edit and delete.*
- ***Databases:***
  - *Example: MySQL, Microsoft Access, SQL Sever, ..*
  - *Data is stored in databases, and each database contains tables.*
  - *Data can be read, write, and retrieve.*

Files (cont..)

- ***Spreadsheets***
  - *Example: Microsoft Excel, Google Sheets, or Numbers*
  - *Tables can be created to store your data.*
  - *Tables are stored in a file, and file can be saved on your computer's disk drive or cloud.*
  - *Calculations, create graphs, and manipulate of the data can be performed by spreadsheets.*
- ***Compilers***
  - *Convert programming code into machine binary readable code*
  - *Each programming language codes can be store in a file such as .cpp for C++, .java for Java, and .py for python.*
  - *Compilers read files, process the code, and generate an executable file.*

Steps for Using File

1. Open/Create the file
2. Use (read from, write to) the file
3. Close the file

File Names

- File name can be a full pathname to file:

  c:\data\student.txt

  tells compiler exactly where to look .

- File name can also be simple name:

  student.txt

  this must be in the same directory as the program executable, or in the compiler's default directory

Streams

- Stream: A transfer of information in the form of a sequence of bytes
- Input stream: Flow into program
  - Input: A stream that flows from an input device (i.e.: keyboard, file “ disk drive”, network connection) to program (main memory)

- Output stream: Flow out of program
  - Output: A stream that flows from main memory to an output device (i.e.: screen, printer, file “ disk drive”, network connection)

Streams in C++

- Streams in C++ provide a convenient way to perform input and output operations with various devices, such as files, standard input/output, and strings.
  - Input is reading data from a file or user input
  - Output is writing data to a file or the console.

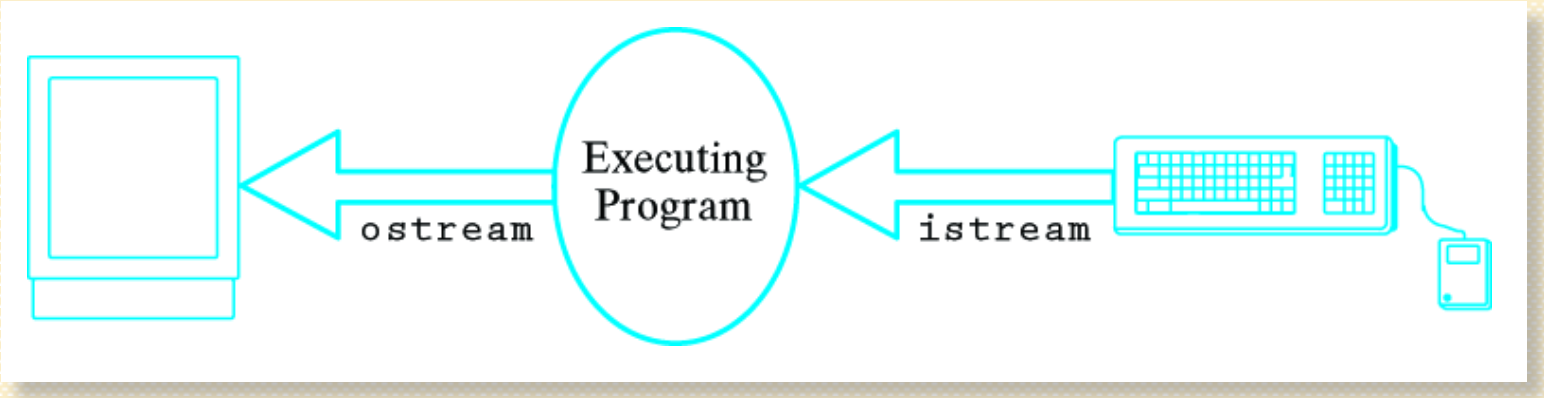
- The <iostream> header provides the necessary classes and functions for stream operations.

- It includes two main stream objects: cin and cout.

  - **cin** is the standard input stream, which is used for reading input from the user via the keyboard.
  - **cout** is the standard output stream, which is used for displaying output to the user on the console.

Streams Usage

- C++ streams are divided into two categories: input streams and output streams.
- An **`istream`** object named **`cin`** connects program and keyboard



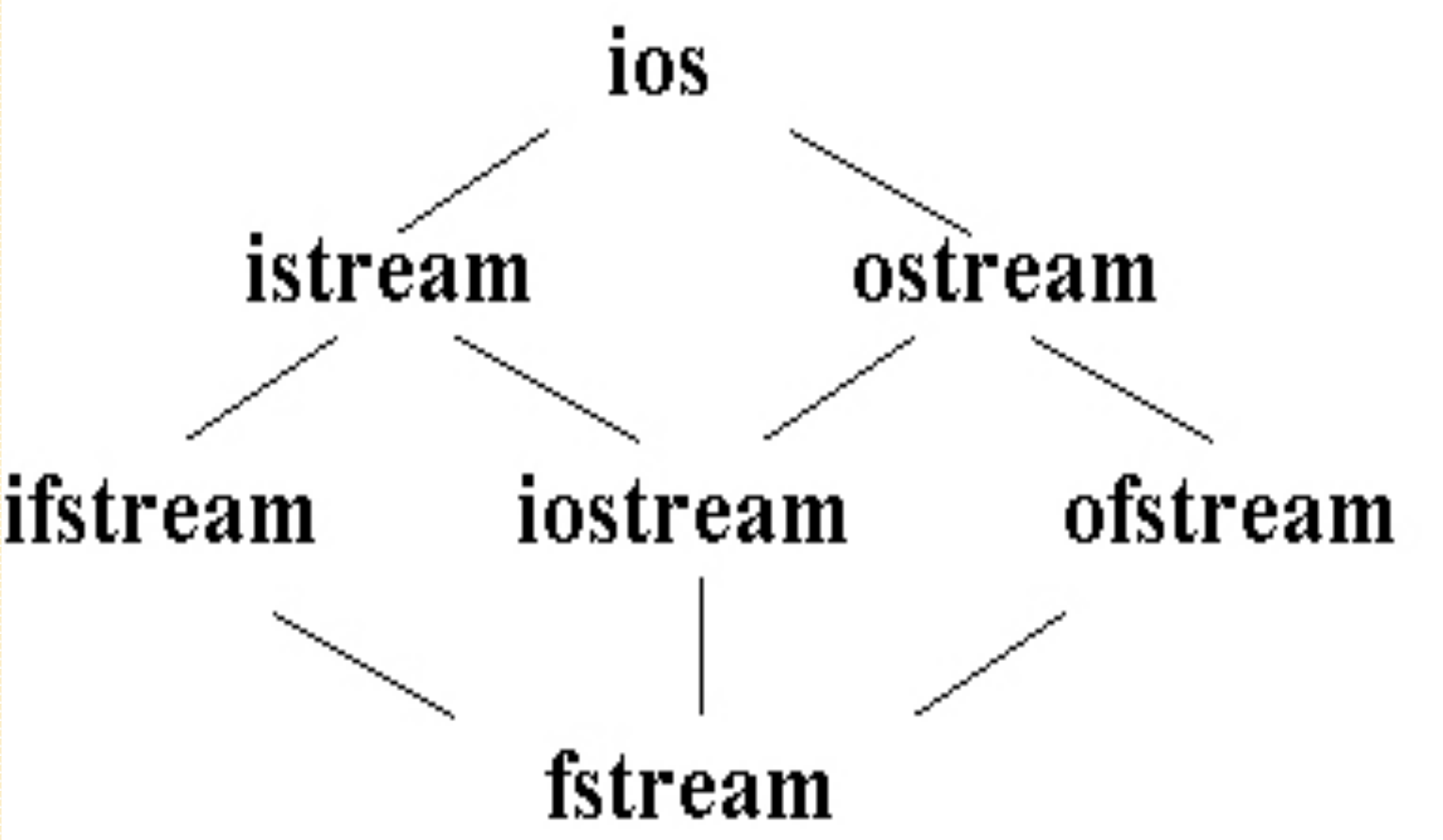
- An **`ostream`** object named **`cout`** connects the program and the screen

Streams Usage

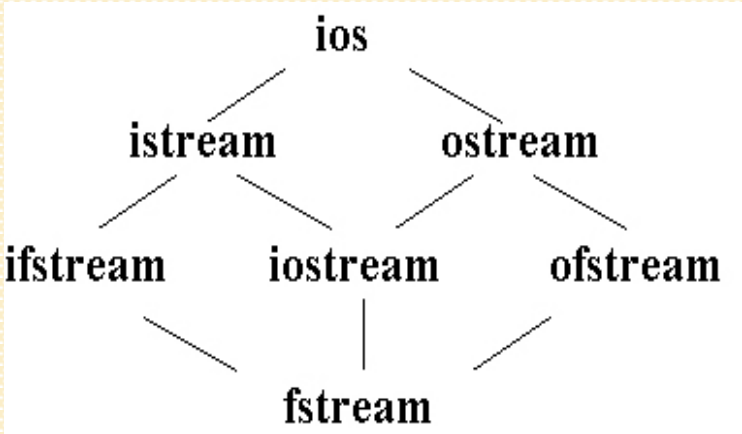
- Input streams (**istream**) are used for reading data from a source, such as a file or the standard input (keyboard).

- Output streams (**ostream**) are used for writing data to a destination, such as a file or the standard output (console).

Classes for Stream I/O in C++



Classes for Stream I/O in C++



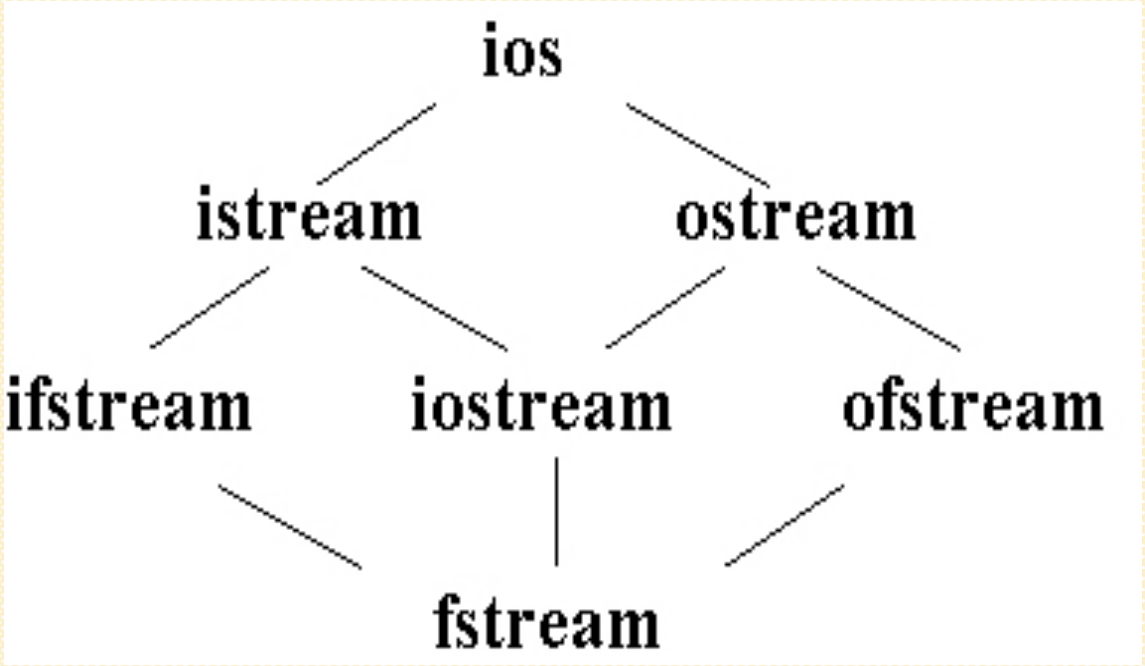
- ios is the base class.
- istream and ostream inherit from ios
- ifstream inherits from istream (and ios)
- ofstream inherits from ostream (and ios)
- iostream inherits from istream and ostream (& ios)
- fstream inherits from ifstream, iostream, and ofstream

File Connection

- Must first connect *file* to *stream object*
- For input:
  - File → ifstream object
- For output:
  - File → ofstream object
- Classes ifstream and ofstream
  - Defined in library <fstream>
  - Named in std namespace

File Stream Objects

- The **ifstream** class is derived from **istream** and provides input operations specifically for file input.
- The **ofstream** class is derived from **ostream** and provides output operations specifically for file output.



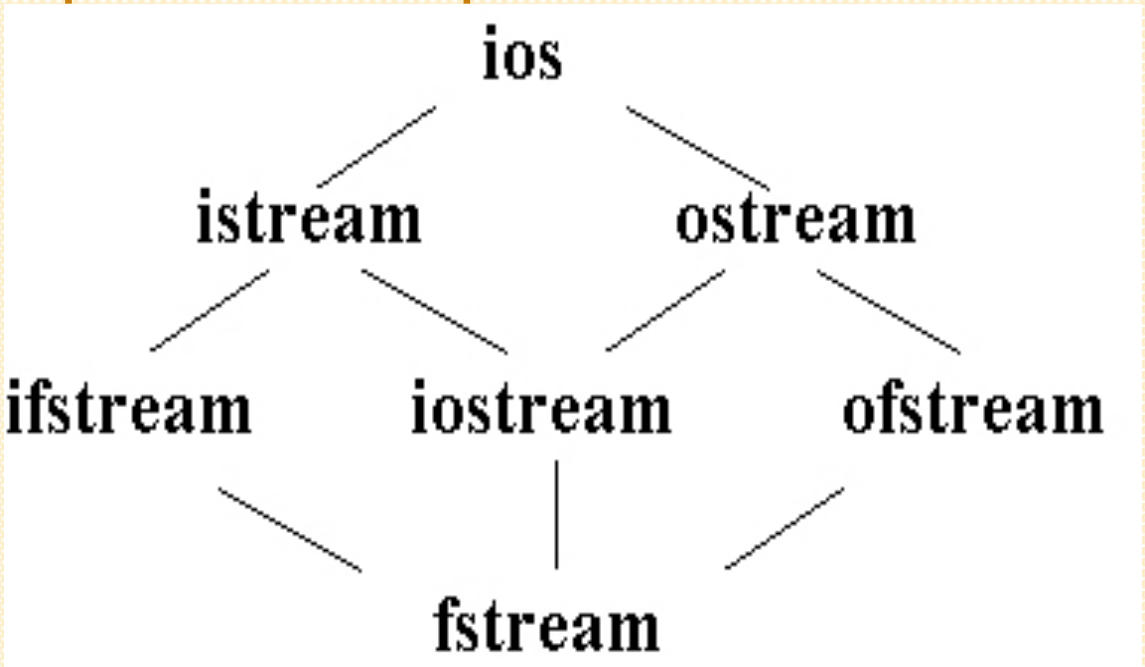
File Stream Objects

- Use of files requires file stream objects
- There are three types of file stream objects

  (1) ifstream objects: input from file

  (2) ofstream objects: out to file or create a file

  (3) fstream objects: used for both input and output



Default File Open Modes

- `ofstream`:
  - *open for output only*
  - *file cannot be read from*
  - *file created if no file exists*
  - *file contents erased if file exists*

- `ifstream`:
  - *open for input only*
  - *file cannot be written to*
  - *open fails if file does not exist*

Closing a File

Traditionally, we close a file when we're done using it:

```
myfile.close();
```

We can do this explicitly, but C++ streams are automatically closed at the end of the variable's lifetime (typically at the end of the function it is declared in)

Declaring Streams

- Stream must be declared like any other class variable:

```
ifstream inStream;
ofstream outStream;
```

- Must then "connect" to file:

```
inStream.open("infile.txt");
```

  - Called "opening the file"
  - Uses member function *open*
  - Can specify complete pathname

Open the file for writing

Opening an input file:

```
ofstream outFile("input.txt");
```

Can also be done in this way:

```
ofstream outFile;
outFile.open("input.txt");
```

Open the file for writing - Example

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    ofstream outFile("firstFile.txt");
    if (outFile.is_open()) {
        outFile << "Hello, Kurdistan!" << endl;
        outFile.close();
        cout << "Data added to the file" << endl;
    } else {
        cout << "Failed to open the file" << endl;
    }
    return 0;
}
```

Open the file for reading

Opening an input file:

```
ifstream myfile("input.txt");
```

Can also be done in this way:

```
ifstream myfile;
myfile.open("input.txt");
```

Open the file for reading - Example

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4  int main() {
5      ifstream inputFile("firstFile.txt");
6      string line;
7      if (inputFile.is_open()){
8          getline(inputFile, line);
9          inputFile.close();
10         cout << line << endl;
11         cout << "File read" << endl;
12     } else {
13         cout << "Cannot open the file" << endl;
14     }
15     return 0;
16 }
```

#include <string>

When you use **getline**, in some programs, you need to add **#include <string>** library

fstream for input and output

- fstream object can be used for both input and output at the same time

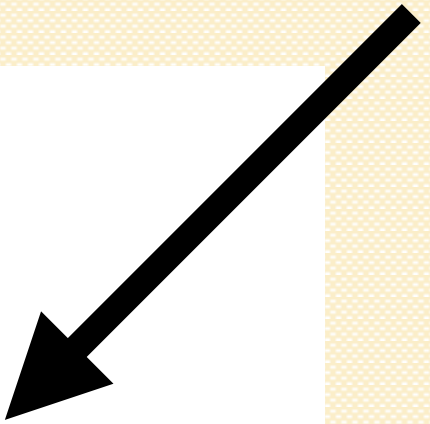
- Create the fstream object and specify both ios::in and ios::out as the second argument to the open member function

```
fstream file; file.open("myfile.txt", iOS::in| iOS::out);
```

Or you can use:

```
fstream file("myfile.txt", ios::in|ios::out);
```

fstream for input and output



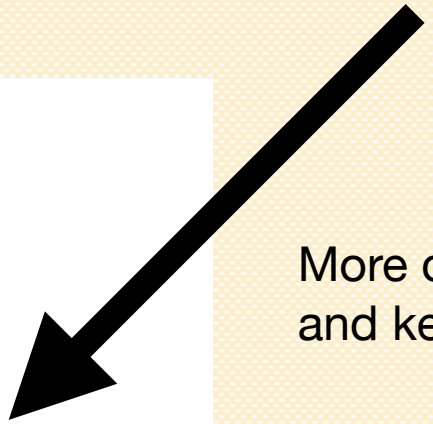
```cpp
#include <iostream>
#include <fstream>
using namespace std;

int main() {
    fstream input("city.txt", ios::in|ios::out);

    if (input.is_open()) {

      input<<"Erbil"<<endl;

     input.close();
     cout<<"Success"<<endl;
    } else {
        cout<<"Failed"<<endl;
    }
    return 0;
}
```

fstream for input and output

`ios::app`	create new file, or **append** to end of existing file
`ios::in`	open for input
`ios::out`	open for output

fstream with ios::app

```
1   #include <iostream>
2   #include <fstream>
3   using namespace std;
4
5   int main() {
6       fstream input("city.txt", ios::app);
7       if (input.is_open()) {
8
9         input<<"Slemani"<<endl;
10
11        input.close();
12        cout<<"Success"<<endl;
13      } else {
14          cout<<"Failed"<<endl;
15      }
16      return 0;
17  }
```



More data to append , and keep the old data

Read some string data from file using while loop

```
1   #include <iostream>
2   #include <fstream>
3   #include <string>
4   using namespace std;
5   int main() {
6       string x;
7       ifstream readFile("city.txt");
8       if (readFile.is_open()) {
9         while (getline(readFile, x)) {
10              cout << x << endl;
11            }
12             readFile.close();
13             cout<<"Read it Successfully"<<endl;
14      } else {
15         cout<<"Failed"<<endl;
16      }
17      return 0;
18  }
```

Read some string data from file using for loop

```
1   #include <iostream>
2   #include <fstream>
3   #include <string>
4   using namespace std;
5   int main() {
6       string x;
7       ifstream readFile("city.txt");
8        if (readFile.is_open()) {
9          for(int i=1; getline(readFile, x);i++){
10              cout <<x << endl;
11           }
12            readFile.close();
13            cout<<"Read it Successfully"<<endl;
14       } else {
15          cout<<"Failed"<<endl;
16       }
17      return 0;
18  }
```

Read some numbers from file using while loop

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

int main() {
    ifstream file("number.txt");
    int number;
    if (file.is_open()) {
      while (file >> number) {
        cout<<number<<endl;
      }
      file.close();
    } else {
      cout<<"Failed"<<endl;
    }

  return 0;
}
```

Read some numbers from file using for loop

```
1   #include <iostream>
2   using namespace std;
3   #include <fstream>
4
5   int main() {
6       ifstream file("number.txt");
7       int number;
8       if (file.is_open()) {
9        for (int i=1; file >> number;i++) {
10           cout<<number<<endl;
11          }
12          file.close();
13      } else {
14       cout<<"Failed"<<endl;
15      }
16
17    return 0;
18  }
```

Read numbers (Same result)

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

int main() {
    ifstream file("number.txt");
    int number;
    if (file.is_open()) {
      while (file >> number) {
        cout<<number<<endl;
      }
      file.close();
    } else {
      cout<<"Failed"<<endl;
    }

  return 0;
}
```

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

int main() {
    ifstream file("number.txt");
    int number;
    if (file.is_open()) {
      for (int i=1; file >> number;i++) {
        cout<<number<<endl;
      }
      file.close();
    } else {
      cout<<"Failed"<<endl;
    }

  return 0;
}
```

Read string (Same result)

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    string x;
    ifstream readFile("city.txt");
     if (readFile.is_open()) {
       while (getline(readFile, x)) {
            cout << x << endl;
        }
         readFile.close();
         cout<<"Read it Successfully"<<endl;
     } else {
        cout<<"Failed"<<endl;
     }
    return 0;
}
```

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    string x;
    ifstream readFile("city.txt");
     if (readFile.is_open()) {
       for(int i=1; getline(readFile, x);i++){
            cout <<x << endl;
        }
         readFile.close();
         cout<<"Read it Successfully"<<endl;
     } else {
        cout<<"Failed"<<endl;
     }
    return 0;
}
```

Insert one word into a file using for loop

```
1   #include <iostream>
2   #include <fstream>
3   using namespace std;
4
5   int main() {
6       ofstream input("city.txt");
7       string name;
8       if (input.is_open()) {
9        for(int i=0; i<4;i++){
10        cout<<"Input city name"<<endl;
11        cin>>name;
12        input<<name<<endl;
13       }
14       input.close();
15       cout<<"Added Successfully"<<endl;
16      } else {
17          cout<<"Failed"<<endl;
18      }
19      return 0;
20  }
```

Insert multiple string into a file using for loop

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

int main() {
    ofstream input("info.txt");
    string name;
    if (input.is_open()) {
     for(int i=0; i<4;i++){
      cout<<"Input your full name"<<endl;
      getline(cin, name);
      input<<name<<endl;
     }
     input.close();
     cout<<"Added Successfully"<<endl;
    } else {
        cout<<"Failed"<<endl;
    }
    return 0;
}
```

How to insert some data into a file using while loop?

Any idea?

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

int main() {
    ofstream input("info.txt");
    string name;
    if (input.is_open()) {
     for(int i=0; i<4;i++){
      cout<<"Input your full name"<<endl;
      getline(cin, name);
      input<<name<<endl;
     }
     input.close();
     cout<<"Added Successfully"<<endl;
    } else {
        cout<<"Failed"<<endl;
    }
    return 0;
}
```

Insert long string into a file using while loop

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

int main() {
    ofstream input("info.txt");
    string name;
    bool flag = true;
    if (input.is_open()) {
     while(flag){
      cout<<"Input your full name (type 'exit' to stop)"<<endl;
      getline(cin, name);
      if (name == "exit") {
            flag = false;
       } else {
          input<<name<<endl;
       }
     }
     input.close();
     cout<<"Added Successfully"<<endl;
    } else {
       cout<<"Failed"<<endl;
    }
    return 0;
}
```

Insert some data into file

```
int a=50;
double b=4.9;
string c ="hello";
myfile << a << b << c << endl;
```

Insert random numbers and find the largest

```cpp
#include <iostream>
using namespace std;
#include <fstream>
int main(){
    srand(time(0));
    ofstream x("random.txt");
    if (x.is_open()){
        for (int i = 0; i < 10; i++){
            int number = rand() % 100;
            x << number << endl;
        }
        x.close();
    } else {
        cout << "Faild" << endl;
    }

    ifstream read("random.txt");
    int largest = 0;
    int number;
    if (read.is_open()){
        while (read >> number){
            if (number > largest){
                largest = number;
            }
        }
    } else {
        cout << "Failed" << endl;
    }
    cout << largest << endl;
}
```

Insert array values into file

```
1   #include <iostream>
2   using namespace std;
3   #include <fstream>
4   int main() {
5       int A[5]={2,5,3,7,8};
6       int size=sizeof(A)/sizeof(A[0]);
7       ofstream x("random.txt");
8       if (x.is_open()) {
9           for(int i=0;i<size;i++){
10           x<<A[i]<<endl;
11          }
12           x.close();
13      } else {
14          cout<<"Faild"<<endl;
15      }
16      return 0;
17  }
```

Sample of input from file word by word

```
1   #include <iostream>
2   #include <fstream>
3   using namespace std;
4   int main() {
5       string x;
6       ifstream readFile("info.txt");
7       if (readFile.is_open()) {
8         while(readFile>>x){
9               cout <<x << endl;
10          }
11           readFile.close();
12           cout<<"Read it Successfully"<<endl;
13      } else {
14          cout<<"Failed"<<endl;
15      }
16      return 0;
17  }
```

Inside the file

```
Tishk International University
Grade one
ProgrogrammingII
Welcome to our university
```

Output

```
Tishk
International
University
Grade
one
ProgrogrammingII
Welcome
to
our
university
```

Sample of input from file line by line

```
1   #include <iostream>
2   #include <fstream>
3   #include <string>
4   using namespace std;
5   int main() {
6       string x;
7       ifstream readFile("info.txt");
8        if (readFile.is_open()) {
9          while(getline(readFile,x)){
10                cout <<x << endl;
11             }
12              readFile.close();
13              cout<<"Read it Successfully"<<endl;
14       } else {
15          cout<<"Failed"<<endl;
16       }
17      return 0;
18  }
```

Inside the file

```
Tishk International University
Grade one
ProgrogrammingII
Welcome to our university
```

Output

```
Tishk International University
Grade one
ProgrogrammingII
Welcome to our university
```

Counting number of lines in the file OR Counting number of words in the file

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    string x;
    ifstream readFile("info.txt");
     if (readFile.is_open()) {
       while(getline(readFile,x)){
            cout <<x << endl;
        }
         readFile.close();
         cout<<"Read it Successfully"<<endl;
     } else {
        cout<<"Failed"<<endl;
     }
    return 0;
}
```

How can we modify these two codes to find the number of words and lines?

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    string x;
    ifstream readFile("info.txt");
     if (readFile.is_open()) {
       while(readFile>>x){
            cout <<x << endl;
        }
         readFile.close();
         cout<<"Read it Successfully"<<endl;
     } else {
        cout<<"Failed"<<endl;
     }
    return 0;
}
```

Counting number of words in the file

```
1   #include <iostream>
2   #include <fstream>
3   #include <string>
4   using namespace std;
5   int main() {
6       string x;
7       int counter=0;
8       ifstream readFile("info.txt");
9        if (readFile.is_open()) {
10         while(readFile>> x){
11               counter++;
12          }
13           readFile.close();
14       } else {
15          cout<<"Failed"<<endl;
16       }
17
18       cout<<"This file has "<<counter<<" words."<<endl;
19      return 0;
20  }
```

Counting number of lines in the file

```
1   #include <iostream>
2   #include <fstream>
3   #include <string>
4   using namespace std;
5   int main() {
6       string x;
7       int counter=0;
8       ifstream readFile("info.txt");
9       if (readFile.is_open()) {
10        while(getline(readFile,x)){
11            counter++;
12         }
13          readFile.close();
14      } else {
15          cout<<"Failed"<<endl;
16      }
17
18      cout<<"This file has "<<counter<<" lines."<<endl;
19      return 0;
20  }
```

How to modify this code to insert all data word by word to a vector and print the vector

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    string x;
    int counter=0;
    ifstream readFile("info.txt");
    if (readFile.is_open()) {
      while(readFile>> x){
          counter++;
       }
        readFile.close();
    } else {
       cout<<"Failed"<<endl;
    }

    cout<<"This file has "<<counter<<" words."<<endl;
    return 0;
}
```

Inside the file

```
Tishk International University
Grade one
ProgrogrammingII
Welcome to our university
```

Output

```
Tishk
International
University
Grade
one
ProgrogrammingII
Welcome
to
our
university
```

Inserting all data word by word to a vector and print the vector

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
using namespace std;
int main() {
    string x;
    vector<string> V;
    ifstream readFile("info.txt");
     if (readFile.is_open()) {
       while(readFile>> x){
            V.push_back(x);
        }
         readFile.close();
     } else {
        cout<<"Failed"<<endl;
     }

     for(int i=0;i<V.size();i++){
       cout<<V[i]<<endl;
     }
    return 0;
}
```

Inside the file

```
Tishk International University
Grade one
ProgrogrammingII
Welcome to our university
```

Output

```
Tishk
International
University
Grade
one
ProgrogrammingII
Welcome
to
our
university
```