Faculty of Computing

CS110: Fundamentals of Computer Programming

Class: BESE-16B

Lab 15: File Handling and Data Persistence

CLO 2	Solve given real-world problem by applying appropriate programming concepts and techniques.
CLO 3	Build a program and associated documentation using appropriate IDE and supplementary tools.

Date: 23rd December 2025

Time: 2:00 pm-5:00 pm

Instructor: Dr. Momina Moetesum

Lab Engineer: Mr. Nadeem Nawaz

Lab 14: User Defined Variables using Structures

Learning Objectives:

After completing this section, you will be able to:

- Read from a file.
- Write to a file.
- Use ifstream functions like seekg(), tellg(), read()
- Use ofstream functions like seekp(), tellp(), write()

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

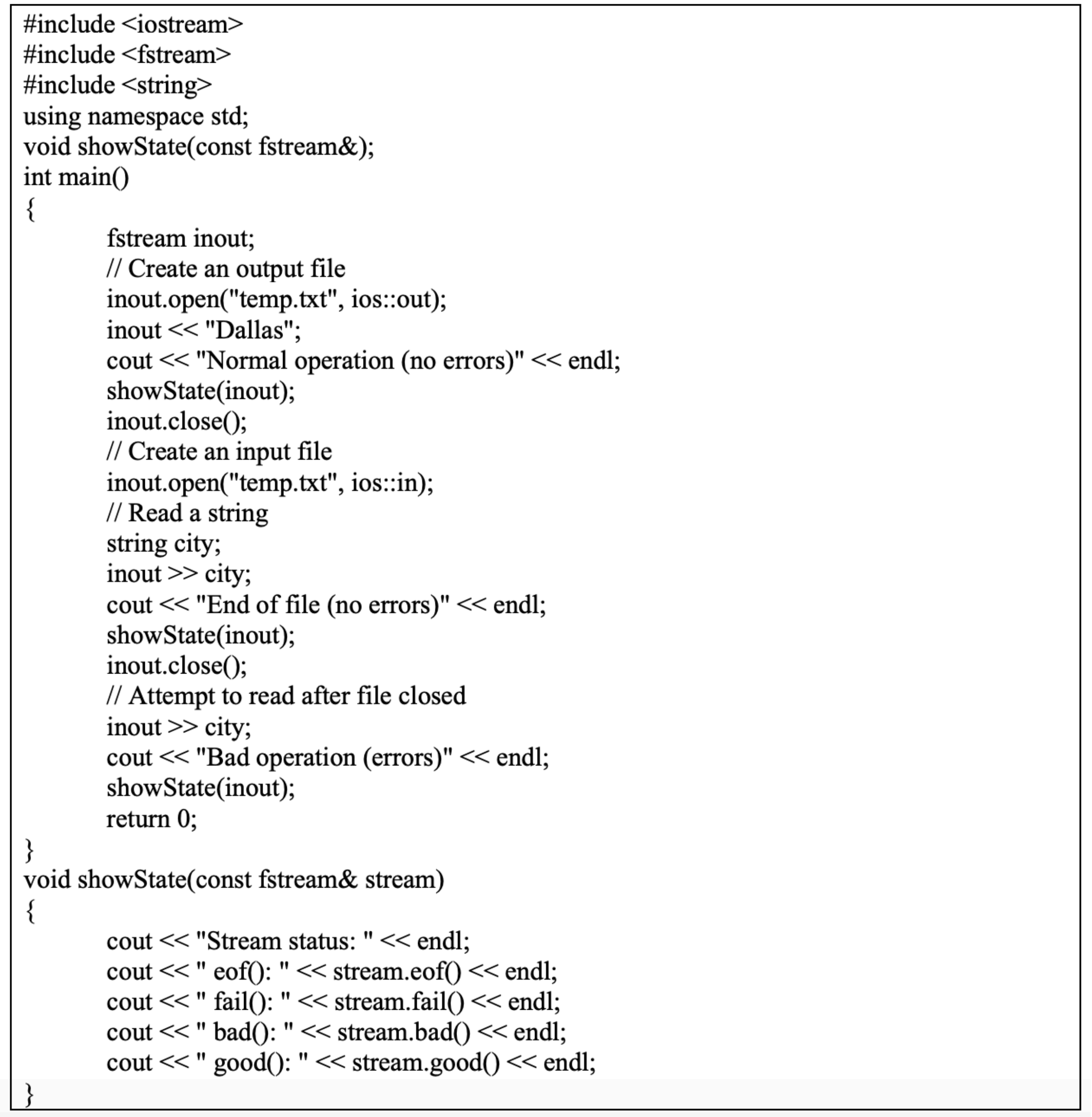
Example Program:

```
#include <iostream>
#include <fstream>
using namespace std;
int main(){
ofstream fout;
fout.open("score.txt");                    // Open a file in write mode
fout<<34<<endl<<45<<endl<<76;              // Write in three lines
fout.close();                              // Close the file
ifstream input("score.txt");               // Open a file in read mode
double sum = 0;
double number;
while (!input.eof())                        // Continue if not end of file
{
   input >> number;                        // Read data
   cout << number <<endl;                   // Display data
   sum += number;
}
input.close();                             // Close the file
cout << "\nSum is " << sum << endl;
return 0;}
```

Lab Tasks

Task 1[CLO3]

While working with streams it is important to know stream status for possible error handling. Run the following program and display the output to see how stream status can be checked in file handling.



```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
void showState(const fstream&);
int main()
{
        fstream inout;
        // Create an output file
        inout.open("temp.txt", ios::out);
        inout << "Dallas";
        cout << "Normal operation (no errors)" << endl;
        showState(inout);
        inout.close();
        // Create an input file
        inout.open("temp.txt", ios::in);
        // Read a string
        string city;
        inout >> city;
        cout << "End of file (no errors)" << endl;
        showState(inout);
        inout.close();
        // Attempt to read after file closed
        inout >> city;
        cout << "Bad operation (errors)" << endl;
        showState(inout);
        return 0;
}
void showState(const fstream& stream)
{
        cout << "Stream status: " << endl;
        cout << " eof(): " << stream.eof() << endl;
        cout << " fail(): " << stream.fail() << endl;
        cout << " bad(): " << stream.bad() << endl;
        cout << " good(): " << stream.good() << endl;
}
```

Task 2: [CLO 2]

Write a C++ program that reads characters from a file if it exists. If it does exist, the program counts the total numbers of characters in the file and prints the count by appending it to the same file.

Task 3: [CLO 2]

Write a program that prompts the user to enter a file name and displays the number of vowels in the file.

Task 4: [CLO 3]

Suppose that a text file contains six integers. Write a program that reads integers from the file and displays their sum, average and product. Integers are separated by blanks.

Deliverables:

Compile a single Word document with codes for each question and screenshots of the outputs and submit this Word file on LMS.

Lab Rubrics

Your Lab 15 will be graded out of 5 for each rubric according to the following rubrics. Grades for CLO3 will be shared at different intervals during the semester.

Lab Rubrics for Lab 15 (File Handling and Data Persistence)

Sr. No.	Assessment	Unacceptable (0 Marks)	Does Not Meet Expectations (1/2 Marks)	Meets Expectations (3/4 Marks)	Exceeds Expectations (5 Marks)
1	**Application of Programming Concepts** **(CLO2, PLO3)**	The student did not submit any work. OR The student plagiarized the solution and/or used unfair means.	The student is unable to apply the appropriate programming concepts to solve the given problem thus resulting in an incomplete or ineffective solution. The program flow is messy and incomprehensible. Codes are non-modular and cannot be reused.	The student requires some guidance to apply the appropriate programming concepts to solve the given problem. The program flow requires minor improvements. Codes are semi-modular and semi-reusable.	The student demonstrates a clear ability to apply the appropriate programming concepts to solve the given problem. The program flow is adequate. Codes are modular, reusable, and easily readable.
2	**Software Tool Usage** **(CLO3-PLO5)**		The student demonstrates a lack of understanding of tool usage. Implementation has syntax/semantic/runtime errors, and the student is unable to debug and correct the errors. The code has inadequate comments and variable names and does not adhere to the coding standards. No Error handling has been performed. Documentation is poorly structured.	The student demonstrates some understanding of tool usage. The codes are correct in terms of their syntax, however, the program output is not always correct in all test cases. The code has limited comments and inconsistent variable names and may not adhere to the coding standards. Some Error handling has been performed. Documentation is adequately structured.	The student demonstrates a good understanding of tool usage. Furthermore, his/her coding is complete and functional, and the program output is correct in all test cases. The code has sufficient comments and consistent variable names and reasonably adhere to the coding standards. Adequate Error handling has been performed. Documentation is well structured.