|  |
| --- |
| **Programming Fundamentals** |
| **(CL214)** |
| **LABORATORY MANUAL** |
| **Spring 2021** |
| **C:\Users\Aamer\Desktop\nu-new.png**  **LAB 07** |
| ***File Handling: Text and Binary Format*** |
| **Engr. Ibrar Khan**  **Engr. Sana Saleh** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_ | | | \_\_\_ |
| STUDENT NAME | | ROLL NO | | | SEC |
|  | | | | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | | |
| LAB ENGINEER SIGNATURE & DATE | | | | | |
| **MARKS AWARDED:** \_\_\_**/10** | | | | | |
|  | | | | | |
| **NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), ISLAMABAD** | | | | | |
|  | | | | | |
| Last Edited by: | Engr. Sana Saleh | | Version: | 2.00 | |
| Prepared by: | Engr. Moomal Bukhari | | Date: | 2 April, 2018 | |
| Verified by: | Engr. Shahid Qureshi | | Date: | 10 March, 2019 | |

|  |  |
| --- | --- |
| **LAB 07** | **File Handling: Text and Binary Format** |

**Lab Objectives:**

1. To learn how to use binary files.
2. To learn how to write to and read from binary files.

**Software Required:**

* Code::Dev C++

**Introduction:**

1. **File Handling**

In C++ file handling library is <fstream>. You create `objects' of type (or class) fstream and use them for reading to and writing from files.

Text files can be manipulated using << and >> operators exactly like cout/cin.

## Text and Binary File Formats

* + The number -3.7182193 can be either a double or float depending on data type being used.
  + For example float f = -3.7182193; is float and takes 4 bytes for storage in memory. These 4 bytes contain the binary representation of float.
  + If stored as a string, it is 10 characters long including the minus sign and decimal point: char c[] = "-3.7182193";. Note: NULL is implicit and doesn't count towards length of string.
  + Take another example of an integer int x = 1;. If stored as a string of characters, it will take only 1 byte. If stored as an integer it will take 4 bytes, regardless of the value.
  + In a text file, data is stored as a string of characters. It's a copy of what you see on your console. As can be expected, a text file is easily readable if opened in a text editor.
  + In a binary file, data is stored as it would in a memory. It's a copy of what is in the memory. A binary file, just like memory, isn't so obviously readable in raw form.

## File Modes

Modes are options that can be specified when opening a file. Options are specified as the second argument of open() function. Options can be combined with the `OR' operator '|'. A few examples are given below:

|  |  |
| --- | --- |
| **Mode** | **Description** |
| ios::in | Open for input operations. |
| ios::out | Open for output operations. |
| ios::binary | Open in binary mode. |
| ios::ate | Set the initial position at the end of the file. If this flag is not set, the initial position is the beginning of the file. |
| ios::app | All output operations are performed at the end of the file, appending the content to the current content of the file. |
| ios::trunc | If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one. |

## Table 1. File Modes

1. **Writing to Text File**

# include <iostream>

# include <fstream > // File handling library .

using namespace std;

int main ()

{

fstream OutFile ; // File handling object .

// Open "abc.txt" for writing ( ios :: out ). This will overwrite existing file

OutFile . open ("abc.txt ", ios :: out );

OutFile << " Hello world ?" << endl ;

OutFile << " Some numbers : 2 3 7 11 13.33 -2.448 " << endl ;

OutFile << " Grades : A A- B+ D C- F A+" << endl ;

OutFile . close (); // Close file

return 0;

}

Code 1. Writing to a Text File

1. **Reading from Text File**

# include <iostream >

# include <fstream > // File handling library .

using namespace std;

int main ()

{

int x;

float f;

char c;

fstream InFile ; // File handling variable .

// Open " abc. txt" for reading ( ios :: in).

InFile . open ("abc .txt ", ios :: in);

InFile >> x >> f >> c;

InFile . close (); // Close file .

cout << " Data read is ... " << endl ;

cout << "x = " << x << ", f = " << f << ", c = " << c << endl ;

return 0;}

Code 2. Reading from a Text File

1. **Writing to Binary File**

First, to write to a binary file, we have to use the binary flag when we create the file:

ofstream outStream("yourFile", ios::binary);

 << is used to write data to a text file. If you had a variable x that contained the value 354 and you used the statement

outfile << x;

This would cause the character 3, the character 5, and the character 4 to be written (in ASCII form) to the file.

The ofstream class provides a member function named write that allows for information to be written in binary form to the stream.

Then, we have to use the write method to write to the file. But this function is really basic and takes only a charand the size of the data we want to write, so we have to convert our data to char. A good way to do that is using the **reinterpret\_cast** function and the **sizeof** operator. For example, to write an int, you can make that:

int test = 22;

outStream.write(reinterpret\_cast<const char\*>(&test), sizeof(int));

Or simply type-cast it using:

outStream.write(( char \*)&test, sizeof(int));

#include <iostream >

#include <fstream > // File handling library .

using namespace std;

int main ()

{ fstream OutFile ; // File handling object . int x = 7; double d = -1.32489; char c = '$ ';

// Open " abc . txt" for writing in binary mode (ios :: out |ios // :: binary ). This will overwrite existing file.

OutFile . open ("abc .txt ", ios :: out |ios :: binary );

OutFile . write (( char \*)&x, sizeof (int ));

OutFile . write (( char \*)&d, sizeof ( double ));

OutFile . write (( char \*)&c, sizeof ( char ));

OutFile . close (); // Close file .

return 0; }

Code 3. Writing to a Binary File

1. **Reading a Binary File**

To read a binary, you have also to open it with the binary tag:

ifstream inStream("yourFile", ios::binary);

Then, you have to use the read method to read your values:

int test;

inStream.read(reinterpret\_cast<char\*>(&test), sizeof(int));

or

inStream.read((char\*)&test, sizeof(int));

# include <iostream >

# include <fstream > // File handling library .

using namespace std;

int main ()

{

int x;

double d;

char c;

fstream InFile ; // File handling object .

// Open " abc . txt" for reading in binary mode (ios :: in|ios :: binary ).

InFile . open ("abc .txt ", ios :: in|ios :: binary );

InFile . read (( char \*)&x, sizeof (int ));

InFile . read (( char \*)&d, sizeof ( double ));

InFile . read (( char \*)&c, sizeof ( char ));

InFile . close (); // Close file .

cout << "x = " << x << endl ;

cout << "d = " << d << endl ;

cout << "c = " << c << endl ;

return 0;

}

Code 4. Reading from a Binary File

## Practice Problem

* 1. A text file is provided to you that contains the data regarding USA women, who obtained a four-year bachelor’s degree from 1970 to 2011. For each year, the percentage of women who obtained a bachelor’s degree in various disciplines/fields is listed. Your tasks for today’s lab are as follow:
     1. Read data from the binary file and store it in a structure named “Yearly\_Data”.
        + Store the 15 disciplines name (first line of the file) in a separate array named “Disciplines”.

(Hint: Store each discipline name in char array of size 50.)

* + - * By using “eof()”, find the total number of years whose is data provided in the file.
      * Use dynamic memory allocation for creating a “Yearly\_Data” array.

**For your convenience, the data is provided in a text file as well. It will help you to know the format.** **Don’t use this file.**

Structure Prototype is as follow:

struct Yearly\_Data

{float Womens\_Percentage[18];

};

* + 1. Print the statistics of graduated women from 1970 to 2011 on the console.
    2. Arrange all the data stored in the structure, in descending order with respect to “Engineering”. Can you tell the year in which the highest percentage of women graduated in the engineering field?
    3. Store this sorted data in text format in a file.
    4. Ask the user about the performance of women in various disciplines in 2019. Append this data in the text file.
    5. Can you find the discipline in which the women outnumbered men in all those years? To find this read data from the text file.

(Hint: Evaluate performance by finding the average of each discipline)

**Extra Tasks:**

1. Using the above data, can you state the years were enrollment of men in “Computer Science” discipline was less than 50%.