CPS235 Object Oriented Programming in C++

LAB10 Introduction to Files and Streams

Objectives

By the end of the lab, you should be able to

- Understand how to open and close a file
- Understand how to write data to a file and read data from a file using the get() and put() functions
- Reading and writing data to files using the stream insertion and extraction operators
- Reading and writing binary data to files using the read() and write() functions

Q1: Write a program which outputs its own C++ source file to the screen.

Q2: Write a program which counts and displays the number of characters (including blanks) in its own source code file.

Q3: Write a program that prints itself out backwards on the screen.
Hint: Write a function that counts the number of characters in the file. Write another function that returns the character at a certain position (given as a parameter to the function) in the file. In main, run a loop which reads the last character in the file, displays it and runs till it reads the first character in the file.

Q4: What screen output does the following program produce, and why?

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

int main()
{
        char character;
        int integer;
        ofstream out_stream;
        ifstream in_stream;

        /* Create a file containing two integers */
        out_stream.open("Integers");
        out_stream << 123 << ' ' << 456;
        out_stream.close();

        /* Attempt to read a character, then an integer,
```

```
        then a character again, then an integer again,
        from the file "Integers" just created.         */
    in_stream.open("Integers");
    in_stream >> character >> integer;
    cout << "character: '" << character << "'\n";
    cout << "integer: " << integer << "\n";
    in_stream >> character >> integer;
    cout << "character: '" << character << "'\n";
    cout << "integer: " << integer << "\n";
    in_stream.close();

    return 0;
}
```

Q5: You are the owner of a hardware store and need to keep an inventory that can tell you what different tools you have, how many of each you have on hand and the cost of each one. Make a class to represent this information. Write a program that initializes the random-access file **hardware.dat** to 100 empty records, lets you input the data concerning each tool, enables you to list all your tools, lets you delete a record for a tool that you no longer have and lets you update any information in the file. The tool identification number should be the record number. Use the following information to start your file:

Record#	Tool name	Quantity	Cost
3	Electric sander	7	57.98
17	Hammer	76	11.99
24	Jig saw	21	11.00
39	Lawn mower	3	79.50
56	Power saw	18	99.99
68	Screwdriver	106	6.99
77	Sledge hammer	11	21.50
83	Wrench	34	7.50

Q6: Convert the banking program in the lecture slides to use a clientData class instead of a structure. The class will have the same data members as the structure whereas the public member functions will include a default constructor, functions to set and get all data members, and a setBalance function to set the value of balance equal to the parameter passed in this function.
In main, write the following functions:

```
void createTextFile( fstream& ); //creates a new file and copies the filled
records in the file passed as argument to a new file create in this function

void updateRecord( fstream& ); // to modify a record in the file associated
with the fstream object passed as argument
```

```
void newRecord( fstream& ); // to add a new record in the file associated
with the fstream object passed as argument

void deleteRecord( fstream& ); // to delete a record in the file associated
with the fstream object passed as argument
```

 void outputLine(ostream&, const ClientData &);// to display the contents of a query on the console using manipulators in the iomanip header file

Acknowledgements