BME506
Fall 2023 – Lab 4
Language Basics – Structs & File I/O

Duration: two weeks.

Note:

1. Every lab assignment must be done individually.
2. When you name a folder or a file, you **should** avoid spaces in those names. For example, if you need to name a folder as **GreenApple**, you should name it as **GreenApple** instead of **Green Apple.** Naming a folder or a file without spaces in the names is to avoid compilation error that may occur while using MinGW build tools (For details, you may see here: http://www.mingw.org/wiki/Getting_Started).
3. **ALL code should be compiled and run on the laboratory computers before submission.**

Goals:

In this lab, you will work with C/C++ programming constructs involving struct, enum, and text file I/O. The task of this lab is related to the construction and archival of transplant patient records. The task includes storage of patient records in an array, display of those records, saving (i.e. writing) of those records in a text file, and finally, loading (i.e. reading) the patient records from a previously saved file into records. The implementation should support up to 100 patients maximum.

Enumerated types will be used for certain patient attributes that have a restricted set of possibilities (e.g. bloodtype), and for file I/O, the following includes should be considered <fstream> and <sstream>. You will need the *getline*() function to grab the entire input string from cin (with whitespaces), and a *stringstream* object to parse it; *ifstream* and *ofstream* objects will be needed to read/write to the text file.

- Please indent your code properly, include appropriate comments, and use proper names for functions and variables.
- System defined header files (if any), specification of namespace (if any), enumerated types (if any), user-defined types (if any), and prototypes of the functions (if any) should be in a header file named **Lab4.h.**
- The implementation of the functions (if any) should be in a file named **Lab4.cpp.**
- The function **main** should also be in **Lab4.cpp.**
- Partial code of **Lab4.h** has already been provided in the **Appendix** at the end of this file. You **should** use it in your implementation.

General steps to build the C++ project in Netbeans for Lab4:

1. Create a Netbeans project named **Lab4Proj** while taking care of the NOTE below.
   NOTE: While creating the project, please **make sure**
   - you set the field "**Project Location:**" appropriately. **Be sure to remember this project location** path. (To keep track of your **Project Location** you may do as follows: Once you have created the project, open a File Explorer, and go into your **Project Location** folder).

- once you have chosen your **Project Location**, enter the word **Lab4Proj** in the field "**Project Name:**". Notice that your project name gets appended to your **Project Location** path.
- you **UN-CHECK** the checkbox on the left of "**Create Main File**" (We do so since we do not need any file named main.cpp file in this project).
- you Click **[Finish]** to create the project.
- a new folder named **`Lab4Proj`** has got created automatically under your **Project Location** folder. This **`Lab4Proj`** folder is your Netbeans ***project folder***.

2. Under netbeans project **Lab4Proj**, right-click on the "Header Files" > "New" > "C++ Header File …". Set the field **"File Name:"** as **`Lab4`**. Set the **"Extension:"** as "**`h`**". Click **[Finish]**.
3. The file **`Lab4.h`** is created. Copy the content for the header file from Appendix into this newly create file.
4. Under netbeans project **Lab4Proj**, right-click on the "Source Files" > "New" > "C++ Source File …". Set the field **"File Name:"** as **`Lab4`**. Set the **"Extension:"** as "**`cpp`**". Click **[Finish]**.
5. The file **`Lab4.cpp`** is created.
6. In **Lab4.cpp**, add the relevant functions that are intended to solve problems of **Part I**, **Part II**, and **Part III** given below.
7. Note: At this point, make sure that your two newly created files **`Lab4.h`** and **`Lab4.cpp`** exist in the **`Lab4Proj`** folder.
8. Right-Click on project **Lab4Proj** >
   Select "Properties" >
   Click "Run" >
   On right, choose "Console Type" to be "Standard Output"
   Click [Apply]
   Click [OK]
9. Build the **Lab4Proj** project.
10. Open a Command Prompt window. Go to the folder **`Lab4Proj\dist\Debug\MinGW-Windows\`** (where **`Lab4Proj`** is the Netbeans *project folder*). You should see that the executable **lab4proj.exe** has got created in this folder.
11. Run the executable **lab4proj.exe** from the command prompt (with proper command-line arguments when necessary).

Important Notes:

1. When your code is error-free, it should build successfully (as seen at the compilation process at Netbean's "Output" window). **However,** at times, in the Netbean's *source editor* window, you may notice red line below a built-in method, such as `size()` method of built-in type string. You may notice this even if your program has built successfully—For example, in the invocation `str1.size()`, you may see a red line below the word `size`. This is misleading. Such a red line should disappear if you perform the following steps:

   Right-Click on project **Lab4Proj** >
   Select "Code Assistance" >
   Select "Clean C/C++ cache and restart IDE"

If you ever want to remove a project from Netbeans, right-click on project > click **"Close". AVOID** using **"Delete"** to remove a project from the Netbeans. If you ever remove a project using "Delete", you will have to create a new project all over again to replace the "deleted" project.

PART I: Use of enum and struct

You are expected to store each patient's information in a struct that records the patient's:

First name;
Last name;
Bloodtype (A, AB, O or B);
Organ (Heart, Kidney, Lung or Liver only);
Age;
Year Added (the year the patient record was added to the list)

** Your implementation should use an array of instances of struct of Patient Record.*
** You **must** use enumerated types (enum) for the attributes (bloodtype and organ).*

Below are five sample patient records:

```
Joe McGurk B kidney 37 2009
Peter Milosevic A heart 28 2012
Alicia Saber O heart 26 2008
John Smith A lung 46 2013
Haylie Jones AB liver 56 2012
```

Each record above has been specified in the following format:

```
<firstName> <surName> <bloodType> <organType> <age> <yearAdded>
```

PART II: Saving/Archiving the Patient Records

You are expected to save the current list of records of patients to a text file **transplantPatients.txt**. The file should be written according to the following format: the first line indicates number of patient records, and each line thereafter represents a new patient record according to the format provided in Part I. The example below shows five sample patient records in the text file **transplantPatients.txt**:

```
Patients 5
Joe McGurk B kidney 37 2009
Peter Milosevic A heart 28 2012
Alicia Saber O heart 26 2008
John Smith A lung 46 2013
Haylie Jones AB liver 56 2012
```

** The file* **transplantPatients.txt** *should get created inside the folder from where* **lab4proj.exe** *is run. Remember to open and close this file with every save! i.e. each save should re-write the entire file.*

PART III: Loading Saved Records

You are expected to implement the loading (i.e. reading) of patient records from the file **transplantPatients.txt** into the array of struct. The executable **lab4proj.exe** should do so when run from the command-line. Once the text file **transplantPatients.txt** has been read, it should be closed.

Example usages of the program for Part I, Part II and Part III are given next.

An *example usage of the program* for Part I and Part II:

```
> lab4proj.exe
======================
[Organ Transplant List]
There are currently 0 patient(s) in the list

Please choose an option:
1. Add Patient
2. Show List of Patients
3. Save List of Patients
4. Quit
>> 1
=============
[Add Patient]
Please enter the Patient Record:

Joe McGurk B Kidney 37 2009
.. [Adding]
======================
[Organ Transplant List]
There are currently 1 patient(s) in the list

Please choose an option:
1. Add Patient
2. Show List of Patients
3. Save List of Patients
4. Quit
>> 1
=============
[Add Patient]
Please enter the Patient Record:

Peter Milosevic A Heart 28 2012
.. [Adding]
======================
[Organ Transplant List]
There are currently 2 patient(s) in the list

Please choose an option:
1. Add Patient
2. Show List of Patients
3. Save List of Patients
4. Quit
>> 2
[Show List of Patients]
ID    Name          Blood   Organ     Age       Year added
-------------------------------------------------------------
1    McGurk, J        B     Kidney     37          2009
2    Milosevic, P     A     Heart      28          2012
======================
[Organ Transplant List]
There are currently 2 patient(s) in the list
```

```
Please choose an option:
1. Add Patient
2. Show List of Patients
3. Save List of Patients
4. Quit
>> 3
[Save List of Patients] (File: transplantPatients.txt)
======================
[Organ Transplant List]
There are currently 2 patient(s) in the list

Please choose an option:
1. Add Patient
2. Show List of Patients
3. Save List of Patients
4. Quit
>> 4
Quitting..
```

An *example usage of the program* for Part III:

```
> lab4proj.exe  transplantPatients.txt
======================
[Organ Transplant List]
There are currently 2 patient(s) in the list

Please choose an option:
1. Add Patient
2. Show List of Patients
3. Save List of Patients
4. Quit
>> 2
[Show List of Patients]
ID    Name          Blood   Organ     Age       Year added
-------------------------------------------------------------
1    McGurk, J        B     Kidney     37          2009
2    Milosevic, P     A     Heart      28          2012
======================
[Organ Transplant List]
There are currently 2 patient(s) in the list

Please choose an option:
1. Add Patient
2. Show List of Patients
3. Save List of Patients
4. Quit
>>
```

Lab Submission

Deadline: See announcement in D2L for deadline.

ALL code should be compiled and run on the laboratory computers before submission.

Create a folder. Name it as YourLastname_YourFirstname_bme506_Labnumber.
Example: If student name is John Smith, the name of folder should be Smith_John_bme506_Lab4.

Copy your Netbeans ***project folder***, **Lab4Proj**, in the above folder. Make sure the folder **Lab4Proj** contains your two files **`Lab4.h`** and **`Lab4.cpp`**.

The above folder must also contain a duly filled and signed standard cover page. The cover page can be found on the departmental web site: Standard Assignment/Lab Cover Page

Compress the above folder as a single zip file that is named according to the following rule:
YourLastname_YourFirstname_bme506_Labnumber.zip.
Example: Smith_John_bme506_Lab4.zip.

Upload the above zip file on D2L through the "Assessment" > "Assignments" link.

Note: If the code does not compile, the submission will receive a ZERO mark.

Resources

[1] Cplusplus.com Reference [Online]. Available: http://www.cplusplus.com/reference
[2] File I/O tutorial [Online]. Available: http://www.cplusplus.com/doc/tutorial/files/

Appendix

In your Lab4.h file, part of the code should be as follows:

```
#ifndef LAB4_H
#define LAB4_H

#include <iostream>
#include <fstream>
#include <sstream>
#include <cstring>
#include <cstdlib>
#include <iomanip>

//WRITE THE REMAINING CODE HERE.
...

#endif /* LAB4_H */
```