BME506
Fall 2023 – Lab 3
Language Basics – Practice with Pointers

Duration: one week.

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 the following lab, you will continue to work with refining your command of core C/C++ programming constructs involving arrays and pointers.

- Please indent your code properly, include appropriate comments, and use proper names for functions and variables.
- The prototypes of the functions should be in a header file named **Lab3.h.**
- The implementation of the functions should be in a file named **Lab3.cpp.**
- The function **main** should also be in **Lab3.cpp.**
- The code of **Lab3.h** and the partial implementation of **main** (that should go in **Lab3.cpp**) has already been provided in the **Appendix** at the end of this file. You **must** use them in your implementation.

General steps to build the C++ project in Netbeans for Lab3:

1. Create a Netbeans project named **Lab3Proj** 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 **Lab3Proj** 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 **`Lab3Proj`** has got created automatically under your **Project Location** folder. This **`Lab3Proj`** folder is your Netbeans ***project folder***.
2. Under netbeans project **Lab3Proj**, right-click on the "Header Files" > "New" > "C++ Header File …". Set the field **"File Name:"** as **`Lab3`**. Set the **"Extension:"** as "**`h`**". Click **[Finish]**.

3. The file **`Lab3.h`** is created. Copy the content for the header file from Appendix into this newly create file.
4. Under netbeans project **Lab3Proj**, right-click on the "Source Files" > "New" > "C++ Source File …". Set the field **"File Name:"** as **`Lab3`**. Set the **"Extension:"** as "**`cpp`**". Click **[Finish]**.
5. The file **`Lab3.cpp`** is created. Copy the content for the source file from Appendix into this newly create file.
6. In **Lab3.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 **`Lab3.h`** and **`Lab3.cpp`** exist in the **`Lab3Proj`** folder.
8. Right-Click on project **Lab3Proj** >
   Select "Properties" >
   Click "Run" >
   On right, choose "Console Type" to be "Standard Output"
   Click [Apply]
   Click [OK]
9. Build the **Lab3Proj** project.
10. Open a Command Prompt window. Go to the folder **`Lab3Proj\dist\Debug\MinGW-Windows\`** (where **`Lab3Proj`** is the Netbeans *project folder*). You should see that the executable **lab3proj.exe** has got created in this folder.
11. Run the executable **lab3proj.exe** from the command prompt.

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 **Lab3Proj** >
   Select "Code Assistance" >
   Select "Clean C/C++ cache and restart IDE"

2. **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: Pointer Arithmetic - Introduction

The following function uses arrays:

```
void accessSalary() {

    int salary[3];

    for (int i=0;i<3;++i) {
        cout <<"Enter Salary: ";
        cin >>salary[i];
    }

    cout << endl;
    cout << "Entered salaries are: " << endl;
    for (int i=0; i<3; ++i) {
        cout << salary[i] << " ";
    }

    cout << endl;
    cout << "Updated salaries are: " << endl;
    for (int i=0; i<3; ++i) {
        salary[i] = salary[i] + salary[i]/(i+1);
        cout << salary[i] << " ";
    }
}
```

Create a new function `accessSalaryArrayByPointerArithmetic`. This new function should modify the above function `accessSalary` as follows: The new function should use a pointer to the array salary, and pointer arithmetic to access and assign values to the array in all the `for` loops (see slides on "Pointers and Arrays" for reference).

Compile the project **Lab3Proj**. Open a Command Prompt window. Go to the folder where the executable **lab3proj.exe** is located. Run the executable from the command prompt (>) as follows:
> **lab3proj.exe part1**

An example output is shown below:

```
> lab3proj.exe   part1

Enter Salary: 111
Enter Salary: 222
Enter Salary: 333

The entered salaries are:
111 222 333
Updated salaries are:
222 333 444
```

PART II: Integer Sort by Pointer Arithmetic

Create a function `sortIntegersByPointerArithmetic` to sort an array of ten integer values in ascending order. The integers should be specified as command-line arguments.

The function should load the integers (passed at command-line) into an integer array.
When `sortIntegersByPointerArithmetic` is called by function `main`, it should:

- accept two arguments: a pointer that points to the loaded integer array, and the array size.
- return a pointer that points to the sorted array.

Note: You **must use pointer arithmetic** to implement the function *sortIntegersByPointerArithmetic*. Also, please indent your code properly, include appropriate comments, and use proper names for functions and variables as required.

[*Hint: you can iterate through the array once, each time considering a swap if the current element and the next element are not in ascending order. Then repeat this n times (where n = number of elements in the array). When you do a swap, use pointer arithmetic (and remember to use temporary storage)*]

Compile the project **Lab3Proj**. Open a Command Prompt window. Go to the folder where the executable **lab3proj.exe** is located. An example run of the executable from the command prompt (>) can be as follows:
> **lab3proj.exe part2 4 3 1 2 9 8 7 6 10 5**

An example output is shown below:

```
> lab3proj.exe  part2  4 3 1 2 9 8 7 6 10 5

Numbers were entered in command-line in the following order:
4  3  1  2  9  8  7  6  10  5
The numbers in ascending order are:
1  2  3  4  5  6  7  8  9  10
```

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_Lab3.

Copy your Netbeans ***project folder***, **Lab3Proj**, in the above folder. Make sure the folder **Lab3Proj** contains your two files **`Lab3.h`** and **`Lab3.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_Lab3.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

Appendix

In your Lab3.h file, the code should be as follows:

```
#ifndef LAB3_H_
#define LAB3_H_
#include <iostream>

using namespace std;

void accessSalaryArrayByPointerArithmetic();
int* sortIntegersByPointerArithmetic(int* arr, int size);

#endif /* LAB3_H_ */
```

In your Lab3.cpp file, the #include statement and the function `main` should be as follows. Fill the body of function `main` as required.

```
#include <iostream>
#include <cstdlib>
#include "Lab3.h"

//implement the function accessSalaryArrayByPointerArithmetic
//implement the function sortIntegersByPointerArithmetic


int main (int argc, char* argv[]) {

    string p1("part1");
    string p2("part2");

    if ( 0 == p1.compare(argv[1]) ) {  //PART I
            //WRITE THE CODE FOR Part I HERE.
            ...

    }

    if ( 0 == p2.compare(argv[1]) ) {  //PART II
            //WRITE THE CODE FOR Part I HERE.
            ...

    }

} //end of main
```