**A Phonebook Application**

You are asked to write a simple C++ phonebook application program. Here are the requirements for the application.

* read the contact information from a given input file (phonebook.txt) into a dynamically created array of Contact objects. Each line of the input line includes name and phone information of a contact. Assume that each name has a single part
* Allow to perform operations on array of data such as search for a person, create a new contact or delete an existing contact

**A sample run:**

\*\*\*MY PHONEBOOK APPLICATION\*\*\*

Please choose an operation:

A(Add) | S (Search) | D(Delete) |L(List) |Q(Quit): A

Enter name: MARY SMITH

Enter phone: 5062396

A(Add) | S (Search) | D(Delete) |L(List) |Q(Quit): S

Enter name: MARY SMITH

Phone Number: 5062396

A(Add) | S (Search) | D(Delete) |L(List) |Q(Quit): L

BARBARA BROWN 4059171

ELIZABETH JONES 2736877

LINDA WILLIAMS 3532665

PATRICIA JOHNSON 973437

…

A(Add) | S (Search) | D(Delete) |L(List) |Q(Quit): D

Enter name: LINDA WILLIAMS

A(Add) | S (Search) | D(Delete) |L(List) |Q(Quit): L

BARBARA BROWN 4059171

ELIZABETH JONES 2736877

PATRICIA JOHNSON 973437

…

A(Add) | S (Search) | D(Delete) |L(List) |Q(Quit): Q

**Sample content from phonebook.txt**

PATRICIA JOHNSON 973437

LINDA WILLIAMS 3532665

BARBARA BROWN 4059171

ELIZABETH JONES 2736877

JENNIFER MILLER 3863726

MARIA DAVIS 6297086

SUSAN GARCIA 6063076

MARGARET RODRIGUEZ 350662

DOROTHY WILSON 2829644

LISA MARTINEZ 6299105

NANCY ANDERSON 3441823

KAREN TAYLOR 1459582

BETTY THOMAS 3977357

HELEN HERNANDEZ 6798052

SANDRA MOORE 4951844

DONNA MARTIN 4335417

CAROL JACKSON 1719547

RUTH THOMPSON 5713343

SHARON WHITE 3236192

MICHELLE LOPEZ 6411624

LAURA LEE 1119042

SARAH GONZALEZ 2212564

**Hint: Sample Codes**

*//A sample code for reading from a file:*

*//readfile.cpp*

#include <iostream>

#include <fstream>

using namespace std;

int main(){

int a, b;

ifstream file;

file.open("test.txt");

file>>a>>b;

cout<<a<<b;

return 0;

}

**Content of test.txt**

1 3

*//A sample code for writing to a file:*

*//writefile.cpp*

#include <iostream>

#include <fstream>

using namespace std;

int main(){

int a=10, b=45;

ofstream outfile("out.txt");

outfile<<a<<" "<<b;

return 0;

}

**Content of out.txt**

10 45

**How to Submit**

* You’ll submit your work via **GitHub** Classroom assignment created by your instructor
* You’ll receive a link to your assignment via Assignment tool on Canvas
* After you accept the assignment, you’ll enter the assignment repository on GitHub
* Click “Clone or Download” and **clone** the assignment onto your computer
* After you make changes, **commit** them. Commits are essentially taking a snapshot of your projects.
* At some point you'll want to get the updated version of the assignment back onto GitHub, either so that we can help you with your code, or so that it can be graded. You can do this by performing **push** operation.

**How to Evaluate**

The following rubric describes how your work will be evaluated.

*Correctness (90 points)*

* [90] Program is correct in object oriented design and function; meets specification
* [75] Program output is correct but elements of specification missing, e.g. variable/method declarations.

[45] Part of the specification has been implemented, e.g. one out of two required

* subprograms.
* [20] Program has elements of correct code but does not assemble/compile.

*Readability (20 points)*

* [10] Programmer name and assignment present. Sufficient comments to illustrate program logic. Well-chosen identifiers.
* [7] Programmer name present, most sections have comments. Fair choice of identifiers
* [5] Few comments, non-meaningful identifiers
* [0] No programmer name. No comments. Poor identifiers