

EECS 1510 Object Oriented Programming

Project 7 – Phonebook 2018

130 Points Due in class Tuesday April 10, 2018

Note 1: This project and the next project may be done in pairs. If working as a pair, submit one copy only, with both names clearly visible. The pairs CANNOT be the same on both projects.

Note 2: There is no extra credit on this project, but considerable extra credit on the next project. The next project is the final project, and is also worth considerably more points.

Consider a program to enter codes of one to eight characters along with an associated telephone number and associated notes. A code can represent a person's name, a person's initials, a place, or anything.

By using such a program, we could enter telephone numbers or notes and associate them with some brief code (i.e. string), which hopefully would be easy to remember. We could then retrieve a given telephone number or notes by entering the code. This might be a useful application for a simple cell phone.

Codes are entered as 1 to 8 characters.
Use "e" for enter, "f" for find, "l" for list, "q" for quit.

```
Command: e Bill
Enter number: 419-536-1234
Enter notes:

Command: e JB
Enter number: 510-0114
Enter notes: Charlie's place

Command: e Jones
Enter number: 413-257-1234
Enter notes: Karen and Jason

Command: e wm
Enter number: 419-257-1234
Enter notes: Walmart

Command: f JB
-- JB
-- 617-510-0114
-- Charlie's place

Command: f Jane
** No entry with code Jane

Command: . . .
```

The "l" command will list all entries on the screen.

For the input to this program, upper and lower case letters are considered equivalent. For example, if a telephone number is entered with the code "Jones", then the codes "JONES", "jones", and "JONes" will all retrieve the telephone number entered with "Jones".

The entries are to be stored in a file from run to run. When the program begins, the entries in the file are to be read into an array. The array should allow for up to 200 entries. The phonebook need not be kept in alphabetical order. You may use a simple sequential search to retrieve entries. When the program is exited, the entries should be stored back in a file for use when the program is run again.

Required Program Characteristics: The assignment is to write a program that incorporates the above features. The entries in the phonebook are to be represented with a simple class:

```
class Entry {
    public String name, number, notes;
}
```

Use an array to store the entries. The array should allow for up to 200 entries.

```
public Entry[] entryList = new Entry[200];
```

Use a static method to read the entries into the program from a file, and a static method to store the entries back into the file when the program is exited.

```
public static void readPhoneBook (String FileName)
    throws Exception
public static void storePhoneBook (String FileName)
    throws Exception
```

Also, use a static method to list all entries:

```
public static void listAllEntries()
```

Submission: See the posted file "Printed Submission of Projects 2018.doc".

- In the printouts of the sample runs, each of the commands "e", "f", "l" and "q" should be illustrated.
- You must also show that the external file is updated with entries added from a previous run.

Writing to a File

```
public static void WritePhoneBook(String FileName) throws Exception{
    PrintStream P = new PrintStream(FileName);

    for (int i=0; i < num_entries; i++) {
        P.println(entryList[i].name + "\t" +
                entryList [i].number + "\t" +
                entryList [i].notes);
    }
    P.close();
    System.out.println("Phonebook stored.")
}
```