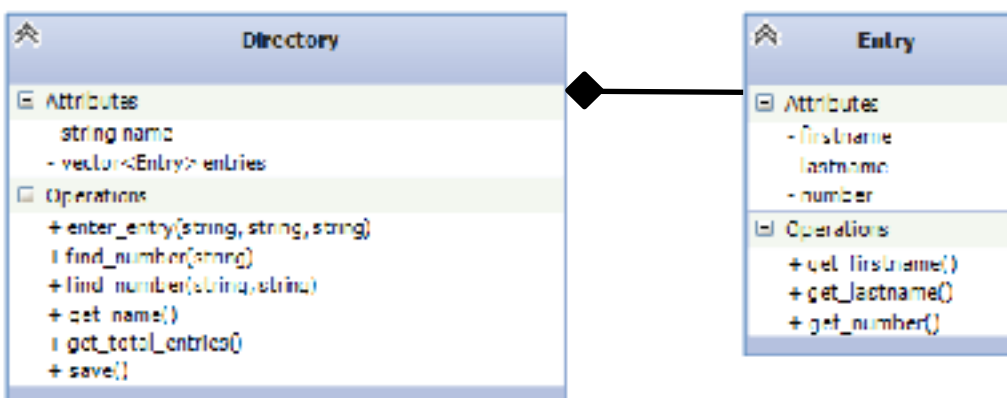


CS172 Exam 2 Practice (Spring 2017)

For this practice, you can look at **ANY** material to complete this. For the actual exam you can only look at your prior programming projects, the book, and your notes. The exam will cover concepts from **chapters 9 to 13**. Please review the power-point slides, our class exercises, and the textbook to prepare for this test.



Figure 1



My telephone directory application defines a **Directory** class to encapsulate all the functionality for my software application. This **Directory** class stores a vector of **Entry** objects, defined by the **Entry** class. The UML diagram describing the **Directory** and **Entry** classes and their relationship is shown in figure 1. (Note the filled diamond symbol represents a “HAS-A” or composition relationship between the classes).

The **Directory** class has TWO properties: 1) a string **name**, describing the name of the directory, 2) a vector of **Entry** objects called **entries**, storing all the telephone directory entries in the application.

The **Entry** class has THREE properties: 1) a string **firstname**, describing the first name of the person in the directory entry, 2) a string **lastname**, describing the last name of the person in the directory entry, and 2) a string **number**, describing the telephone number of the person in the directory entry.

Your task is to implement the **Directory** class in **Directory.cpp** and **Directory.h**, and the **Entry** class in **Entry.cpp** and **Entry.h**.

1. **Constructor**: the constructor initializes the **name** property of the class, and populates the **entries** vector with data read from a file specified by the **filename** parameter. The file contains an unspecified number of lines in the format:

<firstname>, <lastname>, <telephone number>

The format can be examined by looking in the file **TelephoneEntries.txt**.

2. **enter_entry()**: the method adds a new name (specified by a first and last name) and telephone number into the directory. The method ensures that the number entered is in the correct format. If the number format is incorrect, it return **false** and does not add the entry into the directory. Otherwise, it returns **true**, and appends a new **Entry** into the vector **entries**.
3. **get_total_entries()**: the method returns the number of entries currently in the telephone directory.
4. **find_number(string lastname, string firstname)**: the overloaded method returns the telephone number of a person with the specified first and last name. If no directory entry is found, it returns the string **"Not Found!"**.
5. **save()**: saves all the entries (in the **entries** vector) in the telephone directory into the a file called DirectoryOutput.txt. The format of each line in the output file must be as follows:

<lastname> <firstname> <telephone number>

6. **restore()**: restores all of the entries in the DirectoryOutput.txt to the dictionary.

<lastname> <firstname> <telephone number>

Implement also a test program in **main()** that creates a **Directory** object and tests its functionality.