

Homework 1

Submission Details

The solutions for this homework must be submitted as an electronic submission. Your submission will be a tarred and gzipped file containing the source files for the programming problem. These files must be in a directory entitled `HW1_LAST_FIRST` where `LAST` is your last or family name and `FIRST` is your first or given name. Your program will be run to evaluate its correctness, and the source code will be reviewed for adherence to Qt programming style described on the course web site. Your program must run on the latest version of Ubuntu and be compiled/built using the GNU C/C++ compiler and `qmake`. The following information must be included in your source files: your full name, your student ID number, your email address, class (ECE 3574), title of the assignment (Homework 1), and date of submission. You must submit the solutions using the "Submission" button on HW 1 assignment page on the Scholar class web site. The file must be given a name of the following form: `HW1_LAST_FIRST.tar.gz` where `LAST` and `FIRST` are as specified above. You can make multiple submissions. Paper and email submissions will not be accepted. All work must be submitted by the announced due date/time.

Honor Code

As stated in the syllabus, in working on homework and projects, discussion and cooperative learning are allowed. However, copying or otherwise using another person's detailed solutions to assigned problems is an honor code violation. See syllabus for details.

Homework 1 Specification:

For this homework you should write a billing application called "myBillings". This application manages the billings for a consulting business. The interface to this program is from the command line. The program should make use of the Qt classes `QDate`, `QTime`, `QFile`, `QString`, `QStringList`, and `QTextStream`. The program should read and write the billings to a file entitled "mybillings.dat". The format for data stored in this file is up to you, but you must specify the format for this file somewhere in your code comments. Use the example program, `HW1-example`, discussed in class as a starting point.

The format for the arguments on the command line is the following to add a billing:

```
./myBillings -add -n namespec -d datespec -s timespec -e timespec
```

Where the "specs" have the following formats:

- *namespec* is any string contained in quotes, e.g., "Dewey, Cheetem, and Howe". Note that any matching for the name should be case insensitive, so that "Michael Milken" should match "michael milken".
- *datespec* is a date string contained in quotes with the format "mm/dd/yy". For example "01/21/13" or "10/08/12".
- *timespec* is a time in the format "hh:mm". These times are on a 24-hour clock, for example "11:23" or "09:15". The program must check that they the second time is after the first so that the time period makes sense.

If the program successfully adds a billing, the program should print to standard output a string of the form

```
13312 "Dewey, Cheetem, and Howe" 01/11/2013 10:45-11:15
```

The format of the part in blue is up to you; however, the line has to begin with the unique confirmation number (an integer) followed by a space.

The program should also be able to list all billings for a particular day using the following command.

```
./myBillings -list -d datespec
```

Again the billings should be listed, one per line, in the same format as given above for the "add billing" command.

You should also be able to list billings based on the name. The format for this command should be the following.

```
./myBillings -list -n namespec
```

As noted above, the namespec should not be case sensitive, so that "Michael Milken" should match "michael milken".

Given the unique integer corresponding to the confirmation number, your program should be able to delete a billing with the following command:

```
./myBillings -del confnumspec
```

Where the spec has the following format:

- *confnumspec* is the confirmation number in quotes. For example, "13312", where 13312 was the confirmation number returned in the listing described above.

Finally, the command

```
./myBillings -total
```

should print out the total number of hours of billings for each individual client. For example if there were two billings for "Michael Milken", one for 1 hour 30 minutes and one for 2 hours, the total time would 3.5 hours for Michael.

Error Checking:

Of course, you cannot double-bill, bill for work you haven't done yet, or enter other incorrect information. Thus, your program must detect the following errors, and print out an error statement saying what is wrong.

- If the date/time is in the future, the program should return an error.
- If the second time is before the first time, the program should return an error.
- If the billing time overlaps an existing billing time, the program should return an error.
- If command line arguments are incorrect or the specs of the arguments are incorrect, the program should return an error and tell the user the correct format for the commands.

Notes:

- All of your program files should be in one directory, named as specified above.
- The program must correctly compile and link by entering "qmake" and "make" as discussed in class (i.e., you must include a project file in the directory).

Grading Policy

- QT style programming and indentation – 20%.
- The code complies and runs – 0%.
- The code satisfies the specifications and error checking requirements – 80%.