Computer Science 220
Fall 2012
Program 2

Due: Tuesday, September 11, 2012 at 010:50 AM

Learning objectives:

1. Develop a simple Python program that does arithmetic.
2. Develop a Python program with a loop.

Assignment:

Part I

My friend Bubba buys CDs from his favorite store, Esther's Cut-Rate CD Emporium and Plumbing Supplies. For some reason, a lot of CDs show the time of each track, but not the total time of the CD. You are going to help my friend by writing a program that totals the time of each track on a CD. Name your program **`cdTime.py`**.

Your program should first ask the user how many tracks there are on the CD. You may assume the value entered will not be negative, but it could be zero. Your program will now be controlled by a loop that asks for, and adds, the minutes and seconds on each of the tracks. Of course the seconds will frequently add up to more than a minute, but you should not report a total time such as 60 minutes 77 seconds for example (or worse, 55 minutes 377 seconds). Instead you should report the total time as 61 minutes 17 seconds. On the other hand, if the total number of minutes is 60 or greater you should **not** report the time in hours, minutes and seconds. Minutes and seconds are all that are to be reported.

While you may assume that all of the minutes and seconds entered by the user will be non-negative, a time may be entered with seconds that exceed 60. Your program should be able to handle this and still report the total correctly. For example, 5 minutes, 65 seconds and 3 minutes, 88 seconds should add up to 10 minutes, 33 seconds. Consequently for a given CD we should see output such as:

```
Total time: 65 minutes 18 seconds
```

Note: If you already know about conditionals, you **may not** use them in this program.

Part II

Do not try to do Parts I and II all together. Get Part I working completely correctly, and then go on to Part II.

Users of the program report that they do not like having to start the program over and over to process multiple CDs, so we are going to address that complaint.

Add code to first ask how many CDs are to be processed (you may assume the value entered is non-negative). Use this number to determine how many times the program will go through one loop (known as an *outer* loop) that contains within it the code from part 1. Since the loop from the code of part 1 will now lie within the loop you are developing for this part of the assignment, it is known as the *inner* loop. At the end of each pass of the outer loop your program should identify the CD being processed as CD 1, CD 2, etc. and then report the total time of that CD. Thus we may see output such as:

```
CD 1 Total time: 65 minutes 18 seconds
CD 2 Total time: 72 minutes 49 seconds
CD 3 Total time: 49 minutes 1 seconds
```

(It's OK to use the plural form “seconds” in this case.)

Note that the first CD is identified as CD 1, not CD 0.

Finally, add code to output the total time of all of the CDs entered. Thus, the last line of output may be:

```
Total time of all CDs: 3 hours 45 minutes 18 seconds
```

Note that the total time for all CDs **should** display the number of hours, minutes, and seconds.

Documentation and Style: Follow the instructions posted in Policy.doc on OAKS.

Submission: Submit `cdTime.py` to your class OAKS dropbox.