CSCI 1133, Spring 2015
Homework Assignment 1

Assigned: Friday Jan 30, 2015
Due: 11:55pm Thursday Feb 5, 2015

This is ***not*** a collaborative assignment; you must design, implement and test the solution(s) on your own. You may not consult or work with anyone other than the course instructor or TAs. In addition, you may not include solutions or portions of solutions obtained from any source other than those provided in class. Obtaining or *providing* solutions to any homework problems for this class is considered academic misconduct. If you are not sure what this means, consult the class syllabus or discuss it with the course instructor.

Homework assignments involve writing one or more Python scripts that must be submitted online *prior* to the due date/time. Late submissions are not accepted. Name your source code `hw1.py` and submit it via the class Moodle website prior to the stated date/time. Incorrectly named submissions will receive a 10% penalty.

Note that your work will be graded using, and must function correctly with, the current version of Python 3 on CSE Labs UNIX machines. If you complete your programming assignment using a different system, it is your responsibility to ensure your programs work on CSELabs machines *prior* to submitting them. Further instructions for submitting your homework assignment are provided on the class webpage.

A. (20 *points*) **Spare Change**

Write a Python program that will input a monetary (dollar) value and determine the minimum number of coins (quarters, dimes, nickels and pennies) whose total equals the entered value.

For those of you who are unfamiliar with U.S. currency, 1 *dollar* equals 100 *cents*. A *quarter* is worth 25 cents, a *dime* is worth 10 cents, a *nickel* is worth 5 cents and a *penny* is worth 1 cent.

Your program must input two *integer* values representing the dollars and cents, respectively, and then determine and output the minimum number of quarters, dimes, nickels and pennies whose total equals the entered value.

Constraints:

- Use only mathematical operations (no conditional operators)
- Do not import or use any module functions

Examples:

```
Enter dollars: 1
Enter cents: 0
4 quarters
0 dimes
0 nickels
0 pennies

Enter dollars: 2
Enter cents: 42
9 quarters
1 dimes
1 nickels
2 pennies
```