CISC106 – Lab 06

This lab is due 07/21/2013 at 11:55 p.m. (submission via Sakai)

- **Please do all of the following problems in ONE file named lab6.py. This is an INDIVIDUAL assignment, please do all work accordingly.**
- **Use comments to separate your program for each problem. For questions where you should write your answers, envelop them as comments.**
- **For now on we will be using the Design recipe to write functions and the assertEqual function to run our test cases. You need to provide at least 3 test cases for each new function you write, when applicable. Please keep in mind that those elements are worth at least half of the question, so you may not want to forget them!**
- **Make sure the file cisc106.py (attached) is in the same directory of your lab6.py while running it. Also make sure you include it in your lab6.py program. You can do this by adding the following line at the very beginning of your file.**

  from cisc106 import *

- **The problems are worth 90 points + 10 points for attending the lab session.**
- **You do not need to make tests for functions that use random generation (unless they are already provided, on which case you should copy them into your file).**

Problem 1: (10 points)

Implement the sorting algorithm Bubblesort as seen in class through

1) Writing a non-recursive function

2) Writing a recursive function

Provide additional comments explaining which one you thought it was easier to implement and why.

Problem 2: (10 points)

Implement the sorting algorithm Quicksort as seen in class through writing a recursive function.

Problem 3: (10 points)

Implement the sorting algorithm Insertion sort as seen in class through

1) Writing a non-recursive function

2) Writing a recursive function

Provide additional comments explaining which one you thought it was easier to implement and why.

Problem 4: (9 points)

Provide 3 concepts that you would like to represent as classes. Along with them, provide 3 attributes and 3 methods that can be written in order to implement tasks/functionalities of their objects.

Problem 5: (12 points)

1. Provide a model and a class definition for representing points in time since midnight.

2. Write a function that takes two points in time and computes the difference in seconds.

Problem 6: (12 points)

1. Provide a model and class definition for representing three-letter words.

2. Write a function that takes a three-letter word and a character and returns True if the given character is in the three-letter word.

Problem 7: (12 points)

All quadratic equations (in one variable) have the following general shape:

a * x2 + b * x + c = d

1. Provide a model and class definition for representing the LEFT hand side of the quadratic equation.

2. Write a function, **evaluate_quadratic** that takes a quadratic equation and a value for x and calculates the value for d.

Problem 8: (12 points)

Provide a model and class definition for representing students. The information that you should keep about a student is his/her first name, last name, major, GPA and year.

1. Write the method **check_year,** which takes a students and returns the year of the student.

2. Write the method **isComputerScience,** which takes a students and returns True if the major of the student is Computer Science and False otherwise.

3. Create 4 instances of students. You can use these instances as arguments to call and test your methods from the previous items of this question.

Problem 9: (3 points)

Implement requirement 3 of the project (function that checks if the input the user provided is in fact a letter from the alphabet).