ComS 207: Programming I
Homework 10
Out: Wed. Apr 12, 2006
Due: Fri. Apr 21, 2006 (2 a.m. Online submission *ONLY*)

Student Name:

Recitation Section:

Some of these problems came from the new 5-th edition of your textbook. The text is reproduced here for your convenience.

1. **Programming Projects**

   Choose ***three of the following five*** programming projects and implement them. Your grade will NOT depend on which ones you choose. Just pick the ones that you like.

   (a) Shapes Class Hierarchy

   Implement a java class hierarchy that represents at least three different shapes: circles, rectangles, and squares. The base class called **Shape** must have an abstract method called getArea() which is overridden in the derived classes for each shape to produce the correct result. Each derived class must have a constructor which has arguments that are specific for each shape type. For example, each rectangle requires a width and a height while each circle requires only a radius.

   In other words, you must write 5 java classes: Shape.java, Circle.java, Rectangle.java, Square.java, ShapeDemo.java (the driver class). The driver class must instantiate objects of each shape type and demonstrate their functionality (i.e., their ability to calculate their own shape).

   (b) Exam Class Hierarchy

   Implement a java class hierarchy that represents at least three different types of exams: quizes, midterms, and finals. The base class called **Exam** must have three abstract methods called getPoints(), setPoints(), and getNormalizedScore(). In addition to that, each derived class must have a constructor which takes as parameters the minimum number of points for this exam (e.g. 0), the maximum number of points for the exam (e.g., 100), and the weight of this exam (e.g., 20%).

   In other words, you must write 5 java classes: Exam.java, Quiz.java, Midterm.java, Final.java, ExamDemo.java (the driver class). The driver class must instantiate objects of each exam type and demonstrate their functionality (i.e., their ability to calculate normalized scores).

(c) Payable Interface [PP 9.1 in 5-th edition]

Modify the **Firm** example from Chapter 9 such that it accomplishes its polymorphism using an interface called **Payable** instead of overriding the abstract method **pay** declared in StaffMember.java.

(d) Vacation Options [PP 9.2 in 5-th edition]

Modify the **Firm** example from Chapter 9 such that all employees can be given different vacation options depending on their classification. Modify the driver program to demonstrate this new functionality.

(e) Speaker Interface [PP 9.3 in 5-th edition]

Implement the **Speaker** interface described in Section 9.3, and create three classes that implement Speaker in various ways. Create a driver class whose main method instantiates some of these objects and tests their abilities.

2. *** For Advanced (or Bored) Students Only!**

   Linked Lists [Programming project 12.3 (p 633)]

   [THE DESCRIPTION IS IN THE TEXTBOOK.]

3. **What to Submit**

   ONLINE ELECTRONIC SUBMISSION ONLY USING WebCT!!!

   PLEASE, DO NOT SUBMIT PRINTOUTS.

   DEADLINE: 2a.m. on the morning of Friday, April 21.

   IMPORTANT: Once again, no late homeworks will be accepted.

That's it. Good Luck!