**HW 3: Chs 11, 13 and Liang Quizzes**

**Out 10/7/16**

**Due by 10/21/16  
10 points**

**Intermediate Software Development  
SEIS 602.01 - Thursdays**

Fall Semester, 2016

Eric V. Level, Professor

Download and import the zipped Eclipse project **602\_hw\_3.zip** from Class 5 on our Blackboard site.

This project contains starting Java code for the problems below. Follow the instructions below, completing each requested program.

When you are finished, submit by doing **either** of the following:

1. Export your entire project to **602\_hw\_3**, and export it to a .**zip** file. Then email it to me as an attachment,

**OR**

2. Attach each individual **.java** file to an email to me. (Include all files in the project - even those you don't create yourself.) You can find these within the workspace's project folder (same name as project), inside the **src** subfolder.

In either case, put **602 - HW 3** in the Subject: line of your email. Then send it to **evlevel@stthomas.edu.**

Please submit only one entire HW assignment per email. And don't submit both Labs and HW in the same email.

And remember: each student should submit their **own** work for this homework assignment.

Each of the following problems is worth a maximum of 2 points, except for the first which is worth 6 points. Help videos will be posted for each of them under the **Help Videos** section on Bb.

[1] (**p11\_N.P11\_N.java**) Liang's Self-Test Quiz for Chapter 11 asks questions that cover the many mechanisms of Java inheritance. For many of these questions, I created a named package **p11\_*N***, containing a single Java class **P11\_*N***, where ***N*** is the number of the corresponding Chapter 11 Quiz question.

For each such question, I copied its question text, then added it after the **P11\_*N*** class body as a comment. Following this I added a copy of the quiz question's code, formatted it, and then commented it all out. For some of the problems that gave code fragments, I put these in the **main()** method of the **P11\_*N*** class. You will likely want to uncomment this code to figure out the correct answer and understand why it is so.  
  
Pick any 12 of these question classes, then add brief explanations within each of their commented questions. These should explain (a) which answer is correct and why, and (b) why each of the other options is incorrect. Put each of these explanations immediately after its corresponding answer. For those questions you don't do, delete its corresponding package - so that it will be easier to grade. Each of your answers is worth 0.5 points, for a total value of 6 points.

**Extra Credit**: For an 0.5 additional point per problem, do more than the minimum 12 problems.

[2] (**Dedupe.java**) Do Problem 11.14\* at the end of Liang Chapter 11. Put your **union()** method within the **Dedupe** class, with its **main()** method implementing the specified tests. (\*Fixed from earlier version.)

[3] (**Pictures.java**) Using the **Turtle** class from HW 2 and included in the provided starting code for this homework, implement an **abstract** class **TurtleDrawing**. It should have a single **protected** **Turtle** field **turtle**, a constructor **TurtleDrawing(Turtle)** which initializes this field, and an **abstract** method **public void draw()**.

Then define two subclasses of **TurtleDrawing**, where each implements **draw()** so that it draws some interesting picture. Each of these subclasses should have a constructor with a single **Turtle** parameter, storing it in its superclass **Turtle turtle** field, then using this reference field within the **draw()** to draw the picture. Note that your subclasses should NOT create their own **Turtle** objects; instead, a **Turtle** is instantiated outside and passed into the constructors of your **TurtleDrawing** subclasses.

Finally, define the class **Pictures**, with an **ArrayList<TurtleDrawing>** field **pictures**, a method **addPicture(TurtleDrawing td)**, and a method **draw()**. **addPicure(td)** should add a new **TurtleDrawing td** to the **pictures** list, and **draw()** should iterate through this list, calling **draw()** against each of its **TurtleDrawing** objects.

**Pictures.main()** should create one **Turtle** instance, then create an instance of each of your subclasses and pass the same **Turtle** instance into each. This will allow your two picture to share the same drawing area.

**Extra Credit**

From now on, most Extra Credit will be posted in separate documents - and worth 1 point each.