CISC 3120 Fall 2012 Homework 3

Instructions

- This is the first homework assignment for CISC 3120.
- **It is due by midnight on Wednesday September 26th** and you will submit this online — instructions below.
- **Follow these submission instructions:**
  1. Create a ZIP archive from the `.java` files that make up your homework.
  2. Go to:

     `http://agents.sci.brooklyn.cuny.edu/parsons/cisc3120/`
  3. Login and submit your ZIP archive.
  4. You can submit as many times as you like before the deadline—only the final version of the homework will be graded.
  5. Failure to follow these instructions will result in points being taken away from your grade. The number of points will be in proportion to the extent to which you did not follow instructions. . . (since it can make it a lot harder for me to grade your work — grrrr!)

1 Source

You should start from versions of the two Java objects:

`MovingEllipse.java`

`MovingGraphicsComponent.java`

which you can find on the homework page (you used these for homework 1, and many of you have copies from the class this week).

Much of what you will need to do for the homework is what we did in class on Thursday, but make sure read this carefully and do **all** the things this homework asks you to do if you want full credit.

2 First build your project

Once again you you need to use your Eclispe skills to set up a project (previous homeworks describe how to do this in detail).

1. Make sure you include both

   `MovingEllipse.java`

   `MovingGraphicsComponent.java`

   in the project — you will need them before you are done.
2. Remind yourself what this code does — run the project.

3 Class `Point`

1. Write a program that defines the class `Point`. The class should contain two `private` instance fields `x` and `y`.
2. The class should contain `public` API methods to set and retrieve the values of `x` and `y`.
3. The class should contain 3 constructors.

   One should take no arguments and set `x` and `y` to sensible default values.

   One should take two arguments, and set `x` and `y` to these values.

   One should take just one argument, and should use this to set `x` and `y` in some way that you define.
4. Be sure to *document* your program, including writing a comment at the start of each `.java` file that describes the class and the names of the methods that it contains and what those methods do.

4 Using `Point`

1. Modify the `MovingEllipse` project so that the coordinates of the ellipse are held in a `point` object.

   (You will likely only need to modify the file `MovingGraphicsComponent.java`)
2. Make sure you can compile and run this version of the project and that it behaves as before (it can be tricky to make sure you have made all the necessary changes).

5 Class `Eclipse`

1. Write a class `MyEllipse` which has two instance fields: a `point` object and a field that holds the color of the ellipse.
2. The class should have `private` instance fields, `public` API methods, and at least one constructor.
3. Modify the `MovingEllipse` project to use the `MyEllipse` class to hold relevant information about the ellipse that is displayed by the program.
4. Note that you will have to decide how to store the color data in `MyEllipse`, and how to use this to decide what color `paintComponent` invokes.

6 Class `MyTriangle`

1. Write a class `MyTriangle` which has three instance fields all of which are `Point` objects.
2. The class should have `private` instance fields, `public` API methods, and at least one constructor.
3. Modify the `MovingEllipse` project to use an instance of the `MyTriangle` class. The coordinates stored in the three `Points` of the `MyTriangle` should be used to position three ellipses.
4. The data for each of these ellipses should be stored in a `MyEllipse` object.

7 Challenge problems

For extra credit:

1. Figure out how to draw a line between the three points of the `MyTriangle` object, so that a triangle appears on the screen.

2. Allow the user to choose between displaying three ellipses, whose locations are stored in the `MyTriangle` object, and displaying four circles whose locations are stored in a `MySquare` object.

   You will have to create a new class `MySquare` for this.

3. Add some feature to your program that I haven't thought of (you will need to document your code explaining what this is if you want to get credit for it).

8 Document it and hand it in

1. Change the comments in the files:

   `MovingEllipse.java`

   `MovingGraphicsComponent.java`

   to reflect the modifications you made.

2. In particular, you should have a comment that explains how to use your program.

3. This comment should describe any limitations on your code, and any places where your program deviates from the instructions above.

4. You should also document the new classes you wrote, as described above.

5. Submit your work to me, following the instructions above.

9 How I will mark your work

1. You will get credit for each of the things listed above that you have implemented in your program and which work as I have described them.

2. However, your program MUST COMPILE AND RUN for you to get any credit at all.

   (At this point in your computer science career it is unacceptable to submit programs that don't compile and run).

3. Thus it is better for you to submit a program that works and contains less functionality than it is to submit a program with more functionality that does not work.

4. You can get full credit for this homework **without** solving the challenge problems.