Assignment 6

Due: April 1, 2014 at 11:55PM.

You may discuss any of the assignments with your classmates (or anyone else) but all work for all assignments must be entirely your own. Any sharing or copying of assignments will be considered cheating.

You should not use any features of Java that we did not cover in class.

Problem 1: Setup your Eclipse to Run Processing (20 points)

Follow the instructions posted on the website on Feb. 13 to setup your Eclipse to run Processing (direct link to the notes is http://cs.nyu.edu/~joannakl/cs101.06_s14/notes/processingInEclipse.pdf).

Write and run the test program at the end of those instruction and take a screen-shot once the window of the program is running - submit that image for this problem. On my machine it looks something like this:

Problem 2: Car class (80 points)

You are working on a team if programmers designing a new car simulator package. Your job is to write the class that can be used to represent cars in the much larger program.
**Car class description**

The Car type/class should describe the following properties of the car:

- **color** specifies the car’s color; the available colors are blue (B), yellow (Y), red (R), (use character variable to represent it); the default color is red;
- **maxSpeed** specifies the car’s maximum speed in miles per hour; the default value is 150 (i.e. car is not moving);
- **currentSpeed** specifies the car’s current speed in miles per hour; the default value is 0 (i.e. car is not moving);
- **currentPositionX** specifies the car’s current X position in a two-dimensional grid; valid values of X are non-negative integers smaller than or equal to 300; the default X position is 0;
- **currentPositionY** specifies the car’s current Y position in a two-dimensional grid; valid values of Y are non-negative integers smaller than or equal to 300; the default Y position is 0;
- **direction** specifies the current direction of the car; the car can move in one of the four directions indicated by single characters: north ’N’, south ’S’, east ’E’, west ’W’; the default direction is east.

The class should provide the following methods:

- setters and getters for all of the data fields
- constructor that creates a car given its color and maximum speed (all other data fields will be set to their default values) - notice that if either of the variables passed to the constructor is invalid, the constructor should set it to the default value
- constructor that creates a car given its color, maximum speed, current X and Y position (all other data fields will be set to their default values) - notice that if either of the variables passed to the constructor is invalid, the constructor should set it to the default value
- move method that changes the position of the car based on its current position, direction and speed;
  - on each call of this method exactly one hour passes, so the value of either X or Y coordinate should change according to the direction,
  - example: if currentX = 100, currentY = 150, direction = ’N’ and speed = 40mph and the call to move() is made, then the next position should be currentX = 100, currentY = 190 (because 150+40 = 190),
  - if either X or Y moves beyond 300 or becomes smaller than 0, the direction should be reversed to the opposite one and the position set to 300 or 0, respectively.
- toString() method that returns a string containing all the information about the car in the following format:
  
  Car: COLOR, moving DIRECTION at CURRENT_SPEEDmph, located at CURRENT_X, CURRENT_Y

  where all the words in capital letters are replaced by the actual values, for example:

  Car: blue, moving north at 30mph, located at 230, 70

The class should keep track of how many cars are currently in use.

**Driver program**

Write a separate class called Driver. Its main() method should contain the following lines:
Car c1 = new Car ( 'B', 179);
Car c2 = new Car ( 'Z', 200, 100, 150);
Car c3 = new Car ( 'Z', 200, 50, -150);
System.out.println( c1 );
System.out.println( c2 );
System.out.println( c3 );
c1.move();
c2.setDirection('E');
c2.move();
c3.setDirection('W');
c3.setSpeed(70);
c3.move();
System.out.println( c1 );
System.out.println( c2 );
System.out.println( c3 );

Grading

Does the program compile? If not, you will lose all the points for that problem.

Is each class in the program properly documented? - Use Javadoc comments. (worth ~30% of each problem)

Proper documentation includes:

- preamble with the name of the author, date of creation and brief description of the program;
- appropriately chosen variable names, i.e., descriptive names;
- comments inside the code describing steps needed to be taken to accomplish the goal of the program;
- appropriate formatting, indentation and use of white space to make the code readable.

Remember that the code is read by humans and it should be easy to read for people who were not involved in its development.

Is the program well developed? (worth ~40% of each problem) Make sure you create variables of appropriate types, use control statements (conditionals and loops) that are appropriate for the task, accomplish your task in a well designed and simple way (not a convoluted algorithm that happens to produce the correct output for some unknown reason). You should also design a friendly and informative user interface.

Is the program correct? (worth ~30% of each problem). Make sure that your program produces valid results that follow the specification of the problem every time it is run. At this point you can assume a "well behaved user" who enters the type of data that you request. If the program is not completely correct, you get credit proportional to how well it is developed and how close you got it to the completely correct code.

What and how to submit?

You should submit two source code files and an image file combined into a single zip file to NYU Classes. Do not submit all the files that Eclipse creates, just the source code files that have .java extensions. Name your image file Problem1.jpg (or whatever other extension reflects the type of the image); name the two classes Car and Driver (this means that your files are going to be Car.java, Driver.java).