CS 203 HW #3 Movie Poster

Due Date: Wednesday, 05 February 2014, at 10:00pm

You will create a Java program that displays a custom poster for one of your favorite movies or television shows using the Graphics class. This program will give you practice with the Graphics class and if-statements. In addition, you will get some initial experience creating your own methods.

Grading

Your assignment grade will be based on

- Program functionality (80%)
- Code quality (10%)
- Report (10%)

Program Functionality

You will provide the code for two methods in the Art class: `init()` and `paint()`. Additionally, you will create a helper method for use by paint() to draw the a particular element more than once.

`init` method (25%)

Your `init` method should do the following things in this order:

1. A user of your program should be greeted with a dialog window message that introduces the program and asks the user to decide some detail of the poster. (4%) The starter file shows an example of a similar exchange in which the user is asked how many trees to draw. Your program should ask something different. For example, if you were drawing a poster for the Teletubbies, you might ask if one of the Teletubbies should be red or purple.
2. After the user has responded, your program should ask a second question about some other detail of the poster. (4%)
3. After receiving input from each of the dialog windows, the program should verify that the string entered by the user is a valid response. (8%) For example if the user typed in "green" instead of "red" or "purple" in the example above, your program should display a dialog window message stating that the input was invalid and the program is setting the value to a default value. For example, the program might say, "That is an invalid response. The default, 'red', will be used."
4. Each time the user answers a question, you should save the user's response as an instance variable so that other methods in your program will be able to use it later. (5%) Recall that an instance variable is declared in the `Art` class but *outside* of either the `init` method or `paint` method.
5. Any given dialog should only appear once each time the program runs. It should not reappear when the window is resized. (4%)

`paint` method (40%)

Your `paint` method should be implemented as follows:

1. Your `paint` method should display the movie poster in the window for the user. (10%)
2. Your poster should use at least three colors, at least two of which are custom colors (not predefined colors like `Color.red`). (5%)
3. The details of the poster should reflect the choices that the user made. (10%)
4. The elements of the drawing should be recognizable. I don't expect professional graphics, but I should be able to identify the various objects depicted. (10%)
5. Make sure your poster has a title that is drawn using the `drawString` method (5%)

Helper method (15%)

You should implement a helper method that draws a particular element of your poster at a given location.

1. Create a method that, when given a Graphics object and an x,y coordinate (two ints), draws a particular element of the poster at the given position. (10%) For example, if you were doing a Teletubbies poster you might draw a waving Teletubby at the given position. You are welcome to write your method so that it requires additional arguments such as a size or color.
2. It should be possible to run your program so that your `paint()` method calls your helper method at least twice. (5%) It's ok if some user input will cause the method to only be called once or not at all, as long as some user inputs will create two or more calls to your helper method.

Please avoid any elements in your poster that are likely to be offensive to others.

Additional Enrichment (optional)

If you would like to do more for this assignment here are some suggestions.

- A drawing that is better than just recognizable or one that, for some reason, makes the user say "cool!"
- Have some details of the drawing be decided randomly by using the Math.random() method. For example, if you were drawing a garden you might place the plants at random locations.
- Call your helper method in a for-loop to draw several elements of the drawing.

Logistics and Hints

- Download the starter BlueJ project (JavaGraphics.zip) from the course webpage.
- Remember your init() method runs exactly once when the program first begins. So you want to ask <u>all</u> the questions in your init() method. Your paint() method will then display the poster after all the questions have been answered.
- When writing this program, consider getting the `paint` method working with a default configuration before implementing the dialog boxes in the `init` method. That way you

will know that the drawing portion of the program works before you start asking the user to configure it.

- When running the program in BlueJ, you might see the progress bar spin with red and white colors. This means that the program is waiting for user input in a dialog window. If you do not see the dialog, it may be hidden under some other window. (On OS/X, just use exposé.)

Code Quality (up to 10%)

A good computer program not only performs correctly, it also is easy to read and understand:

- A comment at the top of the program includes the name of the program, a brief statement of its purpose, your name, and the date that you finished the program.
- Variables have names that indicate the meaning of the values they hold.
- Code is indented consistently to show the program's structure.
- The body of `if` and `else` clauses are enclosed in braces and indented consistently, even if they consist of only a single line of code.
- Opening braces are placed consistently, either at the end of the if-statement or directly under the 'i' of `if` or the 'e' of `else`. Closing braces are in the same column as the 'i' of `if` or the 'e' of `else`.
- Within the code, major tasks are separated by a blank line and prefaced by one or more single-line comments identifying the task, e.g., "draw a tree."
- Methods are separated by blank lines and prefaced by a multi-line comment describing what they do and what their parameters mean. (See starter code for examples.)
- Very long statements (such as long print statements or complex boolean expressions) are broken across lines and indented to show their structure.

Summary Report (10%)

An effective professional, whether a computer scientist or not, can explain their ideas, their process, and their testing to their coworkers and clients. After you have completed your program, write a brief report about it (no more than 2 pages). When writing the report, assume your audience is someone else taking CS 203. Your report should be contained in a .doc, .docx, .rtf, or .pdf file.

Your report should contain these sections:

- Title: Put a title on your report and include your name as the author.
- Introduction: Describe the program and the tv show or movie that its image is derived from. Why do you like that show? Include a screen shot of a poster drawn by your program.
- Composition: Describe the process you used to compose your poster. Why did you choose the elements that you did? Why did you arrange them as you did?

- Future Expansion: Describe how you could improve your program given additional time. Be specific about the change you would make to the source code so that another CS203 student could use your comments to correctly make the modifications.
- Conclusion: Describe your overall satisfaction with the program and what you learned from completing it.

Turning in this Assignment

You are responsible for turning in your homework assignments properly.

- Be sure your name is in the comment header at the top of your .java file.
- If you did any of the Additional Enrichment (above) carefully document this functionality in the comment header at the top of your .java file.
- Submit your Art.java file (and only this file) via the associated "Turn In Here" link on the course web site. *Do not email your assignment to the instructor.*
- Upload your report via the associated "Turn it in Here" link on the course website. Make sure it is in .doc, .docx, .rtf, or .pdf format. DO NOT turn in a .zip file.