# Final Project Phase I: Create your own picture (Mini Project)

**Description**:  Create a program that draws a picture illustrating some of what we have recently learned about arrays and graphics.   This is an individual project.  You may not work together.  If you have questions or need help debugging, you may ask me or anyone else who is not in the course.

**Requirements**: Your program should create a JPanel within a JFrame window that:
- Includes at least one each of line, oval (or circle), rectangle, polygon and text (in two different fonts)
- Uses a background color other than the default
- Uses at least three different colors of fill
- Uses at least one custom color that is not a predefined Java constant
- Has some elements other than lines that scale with the size of the window
- Contains at least one repeated element with properties controlled by one or more arrays
- Exits when the red x box is pressed.

**Deliverables**:
- The .java file containing your main class and any other classes that you create.
- A screen shot of the picture(s) that your code produces
- A hand drawn, scale diagram on a grid indicating the shapes that make up your drawing along with their initial sizes and positions.

**Assessment Rubric**:

| A. Planning | |
|---|---|
| 0-1 | Design document is missing or incomplete |
| 2-3 | Design document is mostly complete and resembles final product |
| 4-5 | Design document is complete, easy to read, and clearly connected to final product |
| **B. Creativity** | |
| 0-1 | Shows little or no creativity |
| 2-3 | Shows some creativity |
| 4-5 | Shows a lot of creativity |
| **C. Complexity** | |
| 0-1 | Does not meet the minimum requirements |
| 2-3 | Meets the minimum requirements |
| 4-5 | Above and beyond the minimum requirements |
| **D. Coding Style** | |
| 0-1 | Poor formatting<br>Minimal or no documentation<br>Inefficient or incorrect use of code – repetition, hard coded values, etc.<br>Poor choice of variable names |
| 2-3 | Some incorrect formatting<br>Missing documentation<br>Reasonably efficient and mostly correct use of code<br>Mostly appropriate choices for variable names |
| 4-5 | Good use of formatting to make the code readable<br>Complete but concise documentation, including inline comments as appropriate<br>Very efficient and correct use of code – minimal repetition, creative use of methods<br>Thoughtful choices for all variable names |

*Hint*: Start with a simple design that meets the requirements.  Write clean code.  ***Then*** add creativity and complexity.

# Final Project Phase II: Create your own picture

**Description**:  Your final project will be an extension of what we previously called your "Mini-Project" which has since been updated to Final Project Phase I (see above).  The rubric will be the same as previously defined.  However, the final project needs to incorporate the features listed below *in addition to the ones described in Phase I*.

- GUI – Launch your picture from a "GUI" interface that includes at least one button and one text field
- User input/interactivity – some part of your picture needs to be controllable by the user, illustrating three types of user input:
    - Button: Click a button to initiate some visible behavior.  For example, "Press to Start".
    - Keyboard input: Press a keyboard key or combination of keys to initiate some visible behavior.  For example, use the arrow keys to move something like mice running across the screen.
    - Text field: Enter a value in a text field to change some visible behavior.  For example, "Enter # of mice".
- Motion – some part of your animation needs to illustrate motion.

**Due Date**: 11:59 pm, Friday, May 18.

All of the additional features from Phase II will be included in assessment criterion B & C.  You should work out the mechanics of meeting the minimum requirements *before* spending time adding unnecessary complexity.  Design the difficulty of your project to be able to complete the it by the due date.  Your score will be reduced if your work is late without justified extenuating circumstances.