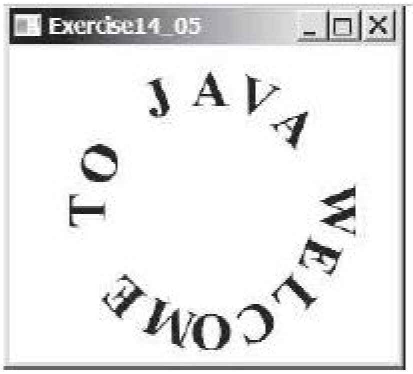
Homework 3
60 points
Due Thursday October 18th 11:59pm

Programming Exercise 14.5 - 20 points

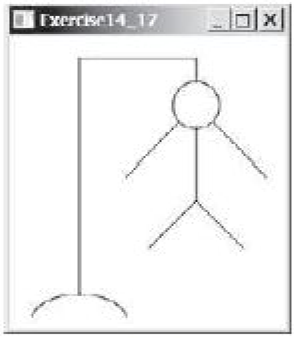
14.5 (*Characters around circle*) Write a program that displays a string Welcome to Java around the circle, as shown in Figure 14.44b. Hint: You need to display each character in the right location with appropriate rotation using a loop.



(b)

Programming Exercise 14.17 - 10 points

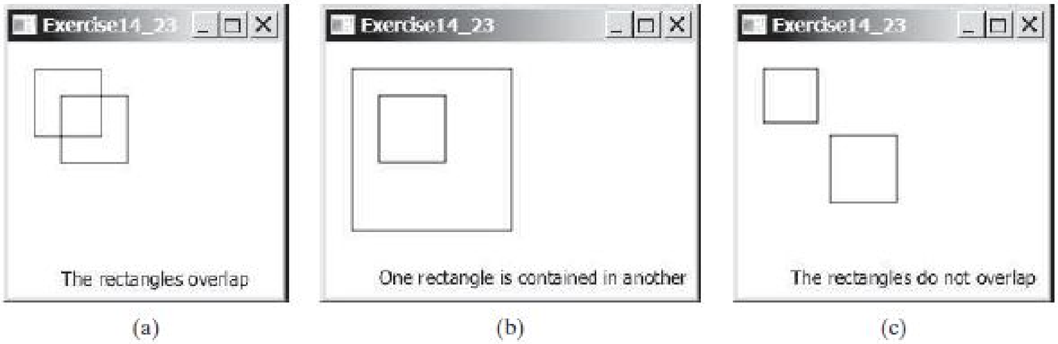
14.17 (*Game: hangman*) Write a program that displays a drawing for the popular hangman game, as shown in Figure 14.48a.



(a)

Programming Exercise 14.23 - 30 points

*14.23 (*Geometry: two rectangles*) Write a program that prompts the user to enter the center coordinates, width, and height of two rectangles from the command line. The program displays the rectangles and a text indicating whether the two are overlapping, whether one is contained in the other, or whether they don't overlap, as shown in Figure 14.50. See Programming Exercise 10.13 for checking the relationship between two rectangles.



The rectangles overlap

(a)

One rectangle is contained in another

(b)

The rectangles do not overlap

(c)

Submit to Canvas:

Submit 1 zip file of all the files in your src folder. Name all your files clearly, so the grader can easily see, and run, which files are for each programming exercise.

Scoring rubric:

The following rubric will be used:

Criteria	**% Points**
Program(s) fulfill all the requirements. All .java files are included and declare the necessary classes. Code is well organized, and easy to follow (especially for the grader). Coding stye is well utilized, including well named variables and methods. Comments are included, well written and descriptive.	100%
Program(s) fulfill almost all of the requirements. All .java files are included and declare the necessary classes. Code is fairly well organized, and somewhat easy to follow (especially for the grader). Some comments are included.	80%

Program(s) fulfill most of the requirements. All .java files are included and declare the necessary classes. Code is fairly well organized, and somewhat easy to follow (especially for the grader). Some or no comments are included.	60%
Program(s) fulfills some of the requirements, or does not run at all. Some .java files are included and declare some of the necessary classes. Some or no comments are included.	40%
Program(s) does not run at all. Some .java files are included and declare some of the necessary classes. Some or no comments are included.	20%
Either no attempt was made, or the attempt made shows no progress toward solving the problem.	0%