

Exercise 1.5: Programming Exercises

Expectations:

1. All code must be commented. Header comments with title, brief description, author, date. Only reasonable in-line code comments. It is not necessary to state the obvious. Using descriptive variables can cut down on the amount of redundant inline comments within your code. Joke commenting will result in joke grading, unless it is relevant or really, really funny (in my view, that is).
 2. Good programming style. Use indentation wisely. Use blank spaces where needed.
 3. Be consistent with your programming style.
 4. Use proper conventions:
 - filename, class name, method names, variable names
 5. Follow established Java programming standards
 6. Do not submit code to me until everything is complete.
-

1. Write a program that reads in integers (as many as the user enters) from the standard input and prints out the maximum and minimum values. (2)
2. Write a program that takes an integer, N , from the command line, reads N double values from the standard input, and prints their mean (average values) and standard deviation (square root of the sum of the squares of their differences from the average, divided by $N-1$) (3)
3. Write a program that reads in a sequence of integers and prints out both the integer that appears in a longest consecutive run and the length of the run. For example, if the input is 1 2 2 1 5 1 1 7 7 7 7 1 1, then your program should print:
Longest run: 4 consecutive 7s (5)
4. Redo question 6, Amortization table, from Exercise 1.3, but use formatting strings to formation your Amortization table. (5)
5. Given the positions and masses of a sequence of objects, write a program to compute their center-of-mass, or centroid. The centroid is the average position of the N objects, weighted by mass. If the positions and masses are given by (x_i, y_i, m_i) , then the centroid (x, y, m) is give by the equations:

$$m = m_1 + m_2 + m_3 + \dots + m_N \quad (5)$$

$$x = (m_1x_1 + \dots + m_Nx_N) / m$$

$$y = (m_1y_1 + \dots + m_Ny_N) / m$$

6. Create a graph using the StdDraw library in stdlib.jar to graph the following equations. (6)
- $y = 5x^3 + 20x^2 + 10x + 100$
 - $y = x^2$
 - $y = 50x^2 + 100$
7. Write a program that takes a command-line argument and plots an N-by-N checkerboard with red and black squares. Color the lower left square red. (5)
8. Write a program that takes a command-line argument N, and plots a rose with N petals (if N is odd) or 2N petals (if N is even), by plotting the polar coordinates (r, θ) of the function $r = \sin(N\theta)$ for θ ranging from 0 to 2π radians.

When plotting points on a Cartesian grid, remember to convert the polar coordinates to Cartesian coordinates using the formula:

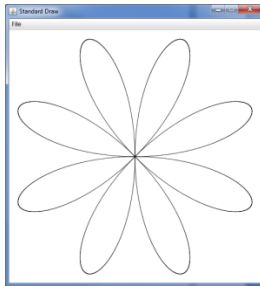
Polar coordinates, (r, θ) , can be converted to Cartesian coordinates, $(r \cos \theta, r \sin \theta)$. (5)

$$x = r \cos \theta$$

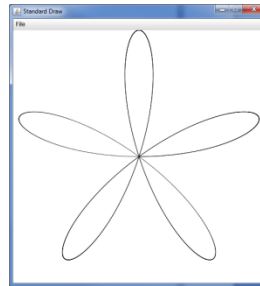
$$y = r \sin \theta$$

Examples:

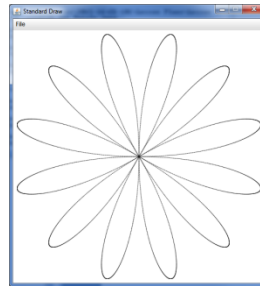
N = 4



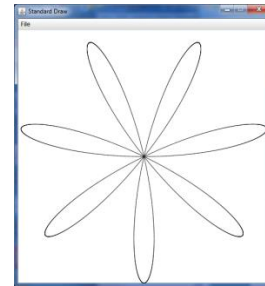
N = 5



N = 6



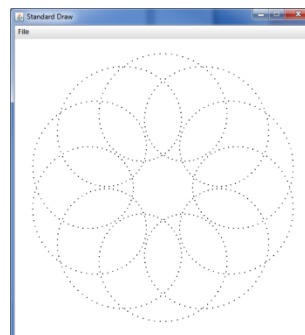
N = 7



9. *Clock*. Write a program that displays an animation of the second, minute and hour hands of an analog clock. Use the method `StdDraw.show(1000)` to update the display roughly once per second. (10)
10. *Spirographs*. Write a program that takes three parameters, R, r, and a, from the command line and draws the resulting spirograph. A spirograph (technically epicycloids) is a curve formed by rolling a circle of radius r around a larger fixed circle of radius R. If the pen offset from the center of the rolling circle is $(r + a)$, then the equation of the resulting curve at time t is given by:

$$x(t) = (R + r) \cos(t) - (r + a) \cos((R + r)t/r)$$

$$y(t) = (R + r) \sin(t) - (r + a) \sin((R + r)t/r)$$



(10)