CS 100: Section Assignment 5

(For the week of March 1)

Section assignments are discussed in section and are not submitted for grading. They relate to recent lecture topics and usually to the current Programming Assignment. Prelim questions are based on Section Assignments, Programming Assignments, and Lecture examples.

This assignment involves the class Disk a listing of which is given below. Here is the relevant geometric background. If D_1 has center (x_1, y_1) and radius r_1 and D_2 has center (x_2, y_2) and radius r_2, then

- Sqrt((x_1- x_2)^2 + (y_1- y_2)^2) is their *center-to-center* distance
- D_1 and D_2 *intersect* if d_{12} <= r_1 + r_2 where d_{12} is the center-to-center distance. If D_1 and D_2 do not intersect, then $d_{12} – r_1 - r_2$ is their *separation.*
- D_1 *contains* D_2 is every point in D_2 is in D_1. This is the case if $d_{12}+r_2 <= r_1$ where d_{12} is the center-to-center distance.

Browse through the class ShowDisk to see examples of how the methods in Disk can be used.

1. Assume that D1, D2, and D3 refer to initialized disk objects. (a) Write a fragment that assigns to a boolean variable B the value true if one of these disks contains the other two. Otherwise the value of false should be assigned to B1. (b) Write a fragment that assigns to a boolean variable B2 the value true if each pair of disks has a positive separation. Otherwise the value of false should be assigned to B2.

2. Add a boolean-valued method inQuadrant() that returns true if the disk does not intersect the *x*-axis or the *y*-axis.

3. Write a while loop that generates random disks. The disks should have radii that are randomly selected from the interval [0,3) and their centers should be randomly selected from the square with vertices (10,10), (-10,10), (-10,-10), and (10,-10). The loop should print all random disks that have a positive separation with the unit radius disk that is centered at the origin. The loop should terminate after five such disks have been printed.

4. If a disk has center (*a*,*b*), then its *reflection* is a disk with the same radius but with center (-*a*,-*b*). Assume that D1 refers to an initialized disk object. Write a fragment that creates a new disk object referenced by D2 that represents its reflection. The fragment should also print the separation between these two disks.