Assignment 3 – Arrays, objects, and GraphicsPrograms

Problem 1 – Array helper methods

The code for this problem should be written in the file `ArrayHelpers.java`.

(a) Fill in the method `findInArray` which takes as arguments a String and an array of Strings, and returns an integer which is the first index in the array at which that String appears, or -1 if that string does not appear in the array. Some examples of how this should function:

- If `arr` is the array {"Hello", "World", "!"}, then the call `findInArray("Hello", arr)` should return 0.
- If `arr` is the array {"green", "blue"}, then the call `findInArray("blue", arr)` should return 1.
- If `arr` is the array {"green", "blue"}, then the call `findInArray("red", arr)` should return -1.

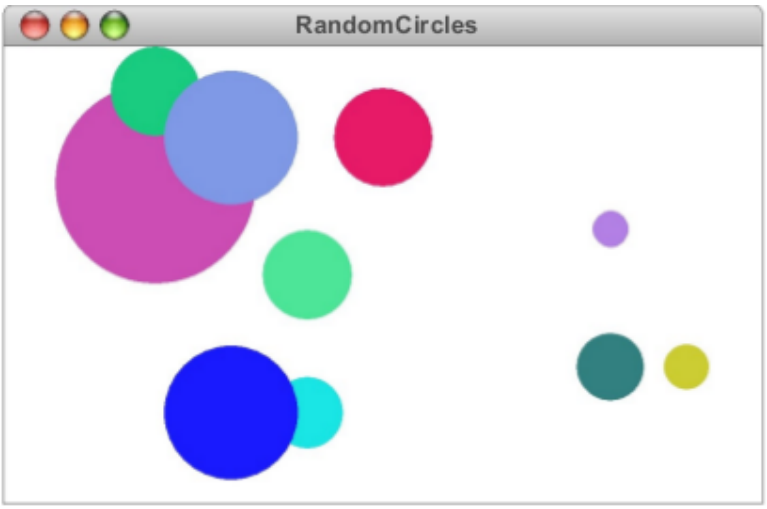
(b) Fill in the method `rangeArray` which takes in an int `limit` and returns an array of integers which contains the numbers 1 through `limit`. For instance, the call `rangeArray(4)` should return an array containing the numbers 1, 2, 3, and 4.

The provided starter code contains several tests of the methods you write. You may feel free to modify these tests or add more as needed.

Problem 2 – Random circles

Your code for this program should go in the file `RandomCircles.java`.

Write a graphics program that draws a set of ten circles with different sizes, positions, and colors. Each circle should have a randomly chosen color, a randomly chosen radius between 5 and 50, and a randomly chosen position on the canvas, subject to the condition that the entire circle must fit inside the canvas without extending past the edge. The following sample run shows one possible outcome:



The provided starter code contains examples of how to generate random numbers and colors in the comments. It also contains several constants, which you should use in your program as appropriate.

Challenge Problem – Sorting an array

In `ArrayHelpers.java`, fill in the method `sortArray`, which takes in an array of integers and modifies it by rearranging its values so that the integers are in sorted order, from smallest to largest.

There are many different ways to sort an array. For one idea of how to do this, look at the Wikipedia page for "bubble sort."