CSCE 145: Algorithmic Design I

Homework - 4

Objective:

Write a program where the user creates and sorts a collection of circles in a variety of ways based on their area.

Requirements:

- Functionality (80 pts)
    - No Syntax, Major Run-Time, or Major Logic Errors. (80 pts*)
        - *Code that cannot be compiled due to syntax errors is non-functional code and will receive no points for this entire section.
        - *Code that cannot be executed or tested due to major run-time or logic errors is nonfunctional code and will receive no points for this entire section.
    - Only use Arrays to store the collection. (80 pts*)
        - *Any other data structure such as LinkedLists, ArrayLists, etc. will result in no points awarded for this entire section.
    - Do not use built-in sorters. (80pts*)
        - *You must implement all sorting yourself.
        - *You MAY NOT use built-in Java sorters such as anything in “import java.util.*” or “import java.util.Arrays”.
        - *Any violation will result in no points awarded for this entire section.
    - Clear and Easy-To-Use Interface (10 pts)
        - Users should easily understand what the program does and how to use it.
        - Users should be prompted for input and should be able to enter data easily.

        - Users should be presented with output after major functions, operations, or calculations.
        - All the above must apply for full credit
    - Provide the user with the following options: (10 pts)
        - Circles' Data: Total # of circles to store in the collection and the radius of each circle in the collection
        - Sort and display the circles' areas from smallest to largest
        - Sort and display the circles' areas from largest to smallest
        - Find and display all unique circles' areas in the collection
        - Quit the program
    - If the user picks an invalid option, then the program must inform the user and continue.
    - All the above must apply for full credit.

- Specify the Size of the Collection and Entering Data. (15 pts)
    - The user must be able to specify the size of the collection (the array). If the size is invalid, then the program must keep asking the user for a valid size. The program should not terminate if an invalid size is entered.
    - After creating the collection, the user must be able to enter the data – the radius of the circle. The radius can be any decimal value: positive, negative, or zero.
    - All the above must apply for full credit.

- Sorting and Displaying the Data. (35 pts)
    - The user must be able to sort the circles by their area either from smallest to largest (ascending order) or from largest to smallest (descending order). Both versions of sort must be implemented for full credit.
    - Area of a circle = πr^2 r → radius of the circle

- The value of π is 3.14
- If the user tries to sort a collection that has not been created (in other words an empty Array), then the program must inform them it cannot sort an empty collection.
- Once the collection has been sorted the areas must be printed to the console.
- All the above must apply for full credit.

- Find and display all unique circles' areas in the collection (20 pts)
  - The user must be able to verify if 2 or more circles have the same radius/area.
    - If all circles in the collection have a distinct values for radius/area, display the areas of all unique circles.
    - If all circles in the collection do NOT have a distinct values for radius/area i.e., some circles in the collection have the same radius/area but some have unique values then,
      - Only display the areas of the unique circles.
      - DO NOT display the areas of the duplicate circles.
  - All the above must apply for full credit.

- Coding Style (10 pts)
  - Readable Code
    - Meaningful identifiers for data and methods.
    - Proper indentation that clearly identifies statements within the body of a class, a method, a branching statement, a loop statement, etc.
    - All the above must apply for full credit.

CSCE 145: Algorithmic Design I

- Comments (10 pts)
    - Your name at the beginning of the file as a single-line comment. (5 pts)
    - At least 5 meaningful comments in addition to your name. These must describe the function of the code it is near. (5 pts)

Submission:

- Submit the .java file on Dropbox