METRO STATE UNIVERSITY

ICS 141 - 02: Problem solving with programming
Spring 2023
Part 2

Assignment 5: Update the “Thing” and Collection class

Out: Wednesday, March, 29, 2023
Due: April, March, 12, 2023 @ 11:59 PM

Total points: 25

In this assignment, you will update your “**Thing**” and collection class to use **compareTo()** and **SelectionSort**.

Requirements

Update your Thing class

In this part of the lab, you will use the selection sort algorithm to sort an array of “Thing” objects.

1- Update your previous Homework #5. Implement the **compareTo** method in your “**Thing**” class as discussed and demonstrated in 03/29 class. Make sure to implement the **compareTo** method to compare objects based on at most two instance variables. For example, if your thing is a Book, then two Books are compared based on year and then on **bookName** (i.e., if two Books have the same year, then they will be sorted on bookName).

2- Copy the **selectionSort** method and paste it in “**Thing**” collection class. Then, change the method so it doesn’t take in any inputs and doesn’t return a value because the collection class already has a reference to an array [] data structure. Now, change the method code to use the **compareTo** method and change any other data types as appropriate.

3- In the main method (“**ThingDriver**”), Call the **selectionSort** method on the “**ThingCollection**”.

4- In the main method, using the existing collection array of any size, fill it with your “**Thing**” objects. Call the **toString()** to print the collection after invoking the **selectionSort** method. Copy the output and paste it here.

Grading: Follow all the steps below to receive full points

Your grade in this assignment is based on the following:

- Your submission meets specifications as described above.
- **Add appropriate comments to your code.**
- Variable names should convey meaning.
- At the top of each Java file, include your name, a brief description of the program and what it does and the due date.
- All code blocks must be indented consistently and correctly. Blocks are delimited by opening and closing curly braces. Opening and closing curly braces must be aligned consistently.
- You must use the exact same name (including upper case and lower-case letter) for all methods as specified in the above description.
- The output of our program must be nicely formatted.
- You must follow the method requirements in terms of the number and data type of input parameters and the output data type.
- The program is robust with no runtime errors or problems.
- The programs should display your name.

Submission Instructions

- At this point, you have completed each section. Part 1 and Part 2.
- Add each Part1 and Part2 solutions into one project folder per the steps below.
- Follow the following steps to upload your code to D2L:
    - Create a java project and call it <yourlastname><your-thing>CollectionWithSelectionSort (e.g., mine using a Student as thing will be called `DillonStudentCollectionWithSelectionSort`)
    - Archive your .java files into **one zip** file using Eclipse using the following steps:
- In Eclipse Project Explorer, right click on the src folder of the project and click on Export.
- Choose General->Archive File and click Next.
- Use the Browse key to choose a folder to store the archive file on your hard drive and give the file the same name as your project (e.g., DillonAssginment5-P2.zip), then click Save, then click Finish.
- Upload the **.zip** file you created to the D2L folder called Assignment 6.
- It is important that you upload only **one** zip file. Your assignment will not be graded if you upload individual .java files to D2L.