**Lab 2**

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name | | Student CSUSM ID | Contribution percentage |
| 1 | Lauren Gonzalez | gonza823 | 50 |
| 2 | Sirena Murphree | murph135 | 50 |

**Grading Rubrics (for instructor only):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | 1. Beginning | 2. Developing | 3. Proficient | 4. Exemplary |
| Modeling | 0-14 | 15-19 | 20-24 | 25-30 |
|  |  |  |  |
| Program: functionality  *correctness* | 0-9 | 10-14 | 15-19 | 20 |
|  |  |  |  |
| Program: functionality  *Behavior Testing* | 0-9 | 10-14 | 15-19 | 20 |
|  |  |  |  |
| Program: quality ->  *Readability* | 0-2 | 3-5 | 6-9 | 10 |
|  |  |  |  |
| Program: quality ->  *Modularity* | 0-2 | 3-5 | 6-9 | 10 |
|  |  |  |  |
| Program: quality ->  *Simplicity* | 0-2 | 3-5 | 6-9 | 10 |
|  |  |  |  |
| Total Grade (100) |  | | | |

**Problems:**

The ABC Company typically uses an object of the SortingUtility class to sort products. A product has at least three attributes: ID, name and price. All are accessible through their corresponding get() method but the ID is fixed once set.

The SortingUtility class implements two private sorting algorithms, bubbleSort and quickSort, each of which takes the list of products and returns an ordered list of products. The SortingUtility class also has a public method List<Product> sort(List<Product> items, int sortingApproach), which simply calls the specified sorting approach (i.e., bubbleSort or quickSort) to return a list of sorted products to its client.

The SortingUtility currently does not log the list of sorted products before returning it to the client. Now the ABC Company would like to have an improved sorting service that can log (for this lab, simply printing to the display console) the list of sorted products before returning it to the client. To implement this improved service you cannot change **the existing** SortingUtility **class for compatible reason**. Moreover, the returned products from bubbleSort should be logged (printed) with ID followed by name and price, whereas the returned products from the quicksort should be logged (printed) with name first followed by ID and price.

(30 pts) What design pattern can be used? Document your pattern-based design in UML class diagram, ensure attributes, methods, visibility, arguments and relationships are correctly included.

(70 pts) Implement your pattern-based design in Java. Implement two test scenarios: one using quicksort to sort a list of products such as books, bags, and buttons, another using bubblesort to sort the same list of products.

**Solution:**

* First, remember to zip the src folder of your project and submit the zip file to the ungraded assignment named “Lab2CodeSubmission”. One submission from each team.
* Paste a screenshot of a run of your program here.
* Also paste all you source code here.
* Save this report in PDF, then **each student** needs to submit the pdf report to the graded assignment named “Lab2ReportSubmission”.