# Code Review Defect List

| Reviewer: \_Jack Northcutt\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | GH Repo: <https://github.com/ntesfai/ntesfai_review> | |  | |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  | |
|  | | |  | |  | |
| **ID #** | **Location** | **Problem Description** | | Problem | | |
| **File and Line Number** | **Category** | | **Severity** |
| **1** | Item.java (entire class) | The class is pointless because it is just creating an Int and assigning the int a value in the constructor | | CS - 5 | | Low |
| 2 | SortDemo.java  line: 23 | SortDemoData data = new SortDemoData(); should be delcared as private and initialized within a constructor. | | CG-5 | | low |
| 3 | SortDemoData.java  lines: 92-120 | Switch case used. | | CS-3 | | low |
| 4 | SortAlgos.java  lines: 274 - 278 | Single line if statements are missing curly brackets | | CG-8c | | low |
| 5. | SortDemo.java  line: 130 | Print statement incorrectly indented | | CG-8b | | Low |
| 6 | SortAlgos.java  line:20 -36 | BubleSort() is incorrectly implemented. The for loops cause the method to skip the first and last element of the array to be sorted. | | FD | | MJ |
| 7 | RandomNumber.java  entire class | SortDemoData uses the RandomNumber class yet imports Java.util.\* which includes the built in random class, so the RandomNumber.java class is redundant. | | CS-1 | | Low |
| 8 | SortDemo.java  lines: 37-137 | Methods: init(),readArray(), displayHelp() and runDemo() all are missing correct method banners | | CG-2 | | Low |
| 9 | SortDemoData.java  lines:3 | Class banner missing description of the class | | CG-1 | | Low |
| 10 | SortDemo.java  line:17 | Imported the entire java.util lib, by using java.util.\* when the class is only using the import for a string tokenizer, wasting overhead. Instead use: java.util.StringTokenizer; | | MD | | low |
|  |  |  | |  | |  |
|  |  |  | |  | |  |
|  |  |  | |  | |  |
|  |  |  | |  | |  |

Category: **CS –** Code Smell defect**. CG –** Violation of a coding guideline. Provide the guideline number. **FD** – Functional defect. Code will not produce the expected result. **MD –** Miscellaneous defect, for all other defects.

Severity: **BR** - Blocker, must be fixed asap. **MJ** – Major, of high importance but not a Blocker **LOW** – Low.