Assignment 4

CSE 214

This assignment is about using different ways of sorting a given set of objects. To illustrate how the same objects can be sorted in different ways, you are required to first implement a class called `Laptop`. Each laptop instantiated by this class should have the following fields: a `String` field called `brand`, a `double` field called `procSpeed` to measure the processor speed in GHz (*e.g.* 3.33), an `int` field called `memory` to measure its RAM, and an `int` field called `hdd` to measure the capacity of its hard disk drive in gigabytes (*e.g.* 500).
Once we have this class ready, we want to sort the laptops according to various features (*e.g.* hard disk capacity or RAM or just by brand name). For this, you must first implement an generic interface called `Sorter` with just two methods:

1. `sort()`
2. `setComparator(Comparator<E> comparator)`

Both methods must return `void`.

The second task is to code a class called `Quicksorter` that implements the `Sorter` interface. This class, too, must be generic. As the name of this class suggests, the `sort()` method must implement the quicksort algorithm.
Because this is a generic class, it is easier to use a list instead of an array. For this, you must use the `java.util.ArrayList` class. For more on arraylists and its methods, take a look at this page:
`http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html`
In particular, you will need to understand `get(int index)` and `set(int index, E element)`.

In this homework, you do not have to write a `main` method. Instead, **your code must be written so that the `main` method given below works.**
<u>DO NOT MODIFY IT</u>, and include it in your `Laptop` class.

```
public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Input data type (must be 'int' or 'laptop'):");
        String inputDataType = scanner.nextLine().trim();
        if (!inputDataType.equals("int") && !inputDataType.equals("laptop"))
            throw new IllegalArgumentException("Invalid data type specified.");

        switch (inputDataType) {
            case "int":
                ArrayList<Integer> integers = readIntegerInputs(scanner);
                Sorter<Integer> intSorter = new Quicksorter<>(intComparator,
                                                              integers);
                intSorter.sort();
                System.out.println(getStringJoinedBy(integers, ", "));
                break;
            case "laptop":
                ArrayList<Laptop> laptops = readLaptopInputs(scanner);
                Sorter<Laptop> laptopSorter = new Quicksorter<>(brandComparator,
                                                                laptops);
                laptopSorter.sort();
                System.out.print("Sorted by brand name:\n\t");
                System.out.println(getStringJoinedBy(laptops, "\n\t"));
                System.out.println();

                laptopSorter.setComparator(processorComparator);
                laptopSorter.sort();
                System.out.print("Sorted by processor speed:\n\t");
                System.out.println(getStringJoinedBy(laptops, "\n\t"));
                System.out.println();

                laptopSorter.setComparator(memoryComparator);
                laptopSorter.sort();
                System.out.print("Sorted by RAM:\n\t");
                System.out.println(getStringJoinedBy(laptops, "\n\t"));
                System.out.println();

                laptopSorter.setComparator(hddComparator);
                laptopSorter.sort();
                System.out.print("Sorted by hard disk capacity:\n\t");
                System.out.println(getStringJoinedBy(laptops, "\n\t"));
                break;
            default:
                throw new IllegalArgumentException("Invalid data type specified.");
        }
    }
```

NOTES

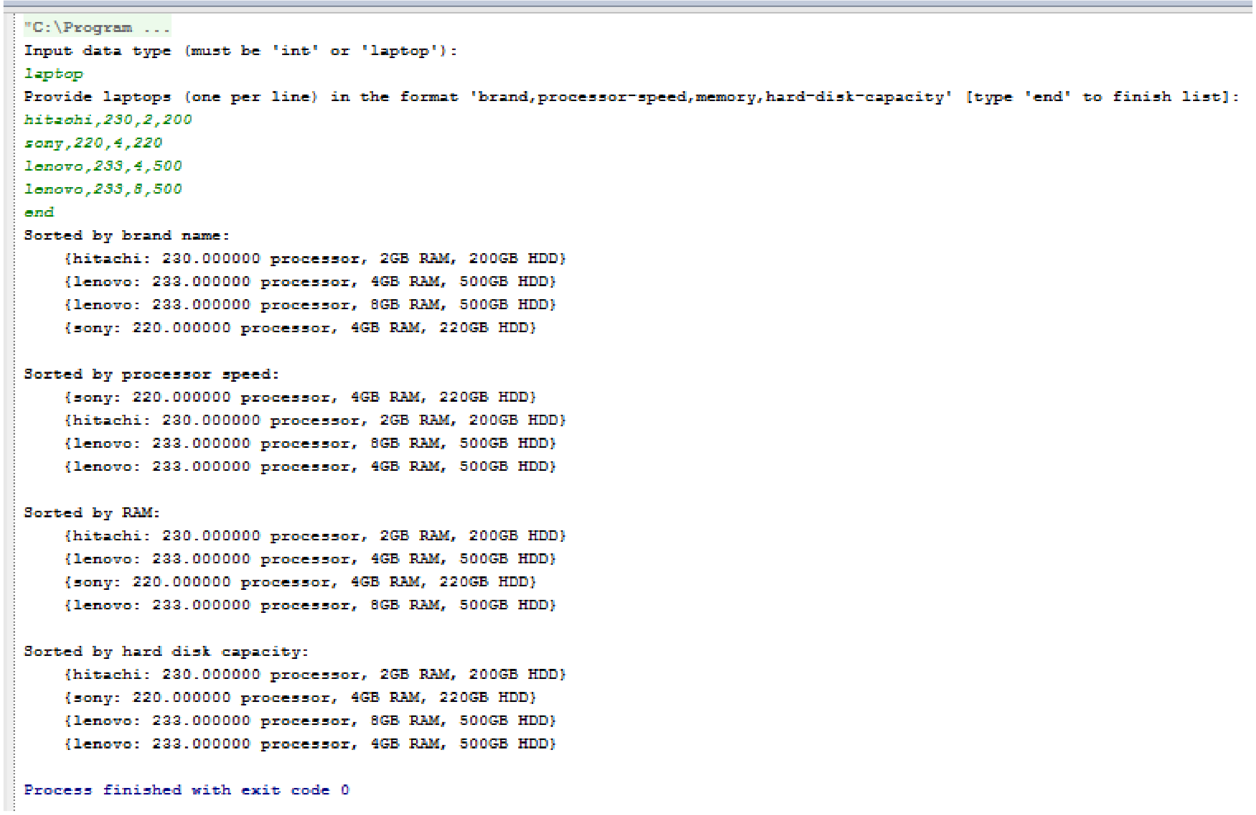
1. You are required to understand how `switch-case` statements work in Java. This IS a part of the syllabus. It is covered in Chapter 6 of the textbook.

2. Just so that it is clear how the user input should be accepted, there are two sample run screenshots in the next page: one for laptops, and one for integers.

3. Some aspects of the class details are intentionally left unspecified. All the necessary details can be inferred from the code and sample runs provided. For example, it uses a method called `readIntegerInputs`, a variable called `hddComparator`, etc. These are things you have to code.

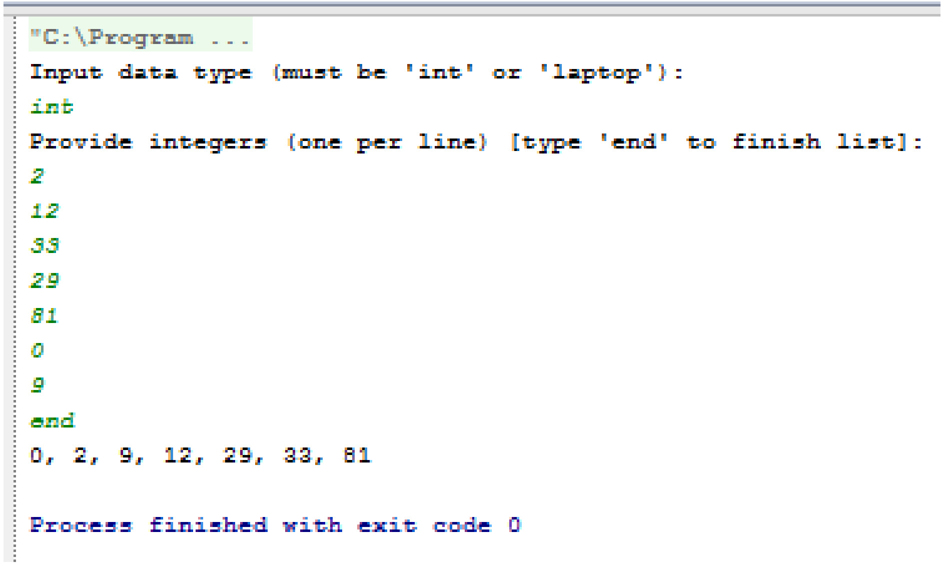
Points Distribution	**Total: 50 points**
1. Defining the sorter interface	2 points
2. Generic implementation of quicksort algorithm	10 points
3. Using comparator for sorting integer input	10 points
4. Using the four field-based comparators for laptop input	28 points

Figure 1: Sample input and output for sorting laptops.



```
"C:\Program ...
Input data type (must be 'int' or 'laptop'):
laptop
Provide laptops (one per line) in the format 'brand,processor-speed,memory,hard-disk-capacity' [type 'end' to finish list]:
hitachi,230,2,200
sony,220,4,220
lenovo,233,4,500
lenovo,233,8,500
end
Sorted by brand name:
    {hitachi: 230.000000 processor, 2GB RAM, 200GB HDD}
    {lenovo: 233.000000 processor, 4GB RAM, 500GB HDD}
    {lenovo: 233.000000 processor, 8GB RAM, 500GB HDD}
    {sony: 220.000000 processor, 4GB RAM, 220GB HDD}

Sorted by processor speed:
    {sony: 220.000000 processor, 4GB RAM, 220GB HDD}
    {hitachi: 230.000000 processor, 2GB RAM, 200GB HDD}
    {lenovo: 233.000000 processor, 8GB RAM, 500GB HDD}
    {lenovo: 233.000000 processor, 4GB RAM, 500GB HDD}

Sorted by RAM:
    {hitachi: 230.000000 processor, 2GB RAM, 200GB HDD}
    {lenovo: 233.000000 processor, 4GB RAM, 500GB HDD}
    {sony: 220.000000 processor, 4GB RAM, 220GB HDD}
    {lenovo: 233.000000 processor, 8GB RAM, 500GB HDD}

Sorted by hard disk capacity:
    {hitachi: 230.000000 processor, 2GB RAM, 200GB HDD}
    {sony: 220.000000 processor, 4GB RAM, 220GB HDD}
    {lenovo: 233.000000 processor, 8GB RAM, 500GB HDD}
    {lenovo: 233.000000 processor, 4GB RAM, 500GB HDD}

Process finished with exit code 0
```

Figure 2: Sample input and output for sorting integers.



```
"C:\Program ...
Input data type (must be 'int' or 'laptop'):
int
Provide integers (one per line) [type 'end' to finish list]:
2
12
33
29
81
0
9
end
0, 2, 9, 12, 29, 33, 81

Process finished with exit code 0
```

Submission Guidelines

1. Submit a single `.zip` file consisting of your `.java` files.
2. Remember to include the proper documentation in your comments in all files!
3. The `.zip` file that you submit should be named as `<SBUID>_<NETID>.cse214.hw4.zip`. For example, Mr. John Doe with student ID number 123456789 will submit the file: `123456789_jdoe.cse214.hw4.zip`.
4. Please make sure that your code compiles and can be run from the command line. **Code that does not compile will not be graded**. In this homework, the main method and sample input/output is already given. So no exceptions will be for uncompilable code.
5. Assignments will **not be graded if you submit the wrong files. No exceptions.** So, **double/triple-check what you are submitting!**

Submission Deadline

The due date for this assignment is midnight of **Friday, Nov 20, 2015**.