Assignment 10%

Individual or 2 person per team :

A group of 2 (maximum) is allowed. No additional marks are given for working alone.

Purpose: Thepurpose of this assignment is to help you review some of the main topics covered in previous courses, includingclasses, loops, arrays, arrays of objects, static attributes and static methods.

General Guidelines When Writing Programs:

- Include the following comments at the top of your source codes

  // .....

  // Assignment (include number)

  // © Your Name

  // Written by: (include your name and student id)

  // .....
- In a comment, give a general explanation of what your program does. As the programming questions get more complex, the explanations will get lengthier.
- Include comments in your program describing the main steps in your program.
- Display clear prompts for users when you are expecting the user to enter data from the keyboard.
- All output should be displayed with clear messages and in an easy to read format.
- End your program with a closing message so that the user knows that the program has terminated.

Part A

A **Computer** object has four attributes, a brand (String), a model (String), an SN (long), and a price (double). SN indicates the serial number of the computer.

For this part, you are required to design and implement the **Computer** class according to the following specifications:

- Upon the creation of a computer object, the object must immediately be initialized withvalid values; that is brand, model, SN and price. (Hint: Constructors.)
- The design should allow enough flexibility so that the value of any of these attributes canbe modified later on. For example, it should be possible to create a computer object with
- a given price then change its price later on. The design should also allow the user toobtain the value of any of the attributes. (Hint: Accessors & Mutators.)
- The design should allow all information of an object to be displayed at once, through theutilization of **System.out.print()** method. (Hint: **toString()** method)
- It is required to know how many Computer objects have been created. For that, you need to add a method, called **findNumberOfCreatedComputers**(), to the class. This method must return the number of created Computer objects prior to the time this method is called. The method would simply return 0 if no computers have been created by the time the method is called. (Hint: You are allowed to add other attributes to the class.)
- It is required to compare two Computer objects for equality. Two Computer objects are considered equal if they have the same brand, model and price. (Hint: **equals()** method)

Part B

You are the owner of a computerstore and need help in keeping track of your computers. Write a driver program that will contain the following methods at least. **Note**: You can have the main function in a separate driver file, or in the same file if you prefer.

1. **a main method**, that will:
    a. Display a welcome message
    b. Prompt the computerstore owner for the maximum number of computers (maxComputers) his/her computerstore can contain. Create an empty array, called **inventory**, that will have the potential of keeping track of the created Computer objects.
    c. Display a main menu (Fig 1) with the following choices, and keep prompting theuser until they enter a number between 1 and 5 inclusive:

```
What do you want to do?
    1. Enter new computers (password required)
    2. Change information of a computer (password required)
    3. Display all computers by a specific brand
    4. Display all computers under a certain a price.
    5. Quit
Please enter your choice >
```

Fig 1. Main menu

    d. When option 1 is entered:
        i. Prompt the computerstore owner for his/her password. (Make sure you have a constant variable containing the password "password" – do not use any other password as it will be easier for the marker to check your assignments). The computerstore owner has 3 tries to enter the correct password. After the 3rd illegal entry, the main menu in figure 1 is re-displayed again.

        ii. If the correct password is entered, ask the owner how many computers he/she wants to enter. Check to make sure that there is enough space in the computerstor (array of **Computer**) to add these many computers. If so, add them; otherwise inform the owner that he/she can only add the number of remaining places in the array. (How the computer information will be input/entered by the user, is up to you).

e. When option 2 is entered :

i. Prompt the computerstore owner for his/her password. (Make sure you havea constant containing the password "password" as a constant – do not use any other password as it will be easier for the marker to check your assignments). Again the computerstore owner has 3 tries to enter the correct password. After the 3rd illegal entry, the main menu in figure 1 is re- displayed again.

ii. Ask the user which computer number he/she000 wishes to update. The **computer** number is the index in the array **inventory**. If there is no **Computer** object at the specified index location, display a message asking the user if he/she wishes to enter another computer, or quit this operation andgo back to the main menu. If the entered index has a valid computer, display the current information of that computer in the following format:

Computer # X
Brand: brand of this
computerModel: model
of computer
SN: serial number (SN) of this
computerPrice: $price

Then ask the user which attribute they wish to change by displaying thefollowing menu.

What information would you like to change?
1. brand
2. model
3. SN
4. price
5. Quit

Enter your choice >

Fig 2. Update menu

Once the user has entered a correct choice, make the changes to the attribute then display again all of the attributes on the screen to show that the attribute has been changed. Keep prompting the user for additional changes until the user enters 5. Each time the user is prompted for a choice make sure that a number from 1 to 5 is entered, otherwise keep prompting until a valid number is entered.

f. When **option 3** is entered, prompt the user to enter a brand name. You then need to display the information of all computers that have that brand. (Hint: You may use a static method, for instance called **findComputersBy** , which accepts a string for a brand name then performs the needed search).

g. When **option 4** is entered, promtp the user to enter a value (representing a price). You then need to display all computers that have a vlaue smaller than that entered value. (Hint: You may use a static method, for instance called **findCheaperThan**, which accepts a double value, for a price, then performs the needed search).

h. When **option 5** is entered, display a closing message and end the driver.

Part II.A (Class Computer)	**4 pts**
Default & copy constructors	1 pt
Accessor/mutator method for static attribute	1 pt
equals, toString and static attributes/methods	2 pts
Part II.B (Driver & other static methods)	**6 pts**
Handling of password	1 pt
Handling of option 1	1 pt
Handling of option 2	1 pt
Handling of option 3	1 pt
Handling of option 4	1 pt
Handling of option 5	1 pt