

# GTU Department of Computer Engineering

## CSE 222/505 - Spring 2021

### Homework 6

**Due date: May 23 2021– 23:55 AM**

Design and implement an e-shopping application that has two user roles as usual: traders and customers. Users authenticate to the system by using their *user IDs* and *passwords*. The user ID is an eight-digit unique number and the password consists of six characters. After the authentication process is done properly, the program shows a menu to the users with respect to their user role. The application meets the following requirements by these menus:

- Traders add, remove, and edit products on their shop. Each product has a unique ID, a hierarchical category, a name, a description, a price, and a discount amount.
- Traders see the list of orders from customers and can meet or cancel the orders.
- Customers can search and query the products by their names. The products that contain the search text in their name or description should be returned in the results.
- The search results are shown in decreasing order of the product name. Customers can sort the products in the search results by their prices, the percentage of the discount, or the name.
- The search results can be filtered by their category or the prices with respect to given upper or lower (or both) thresholds. The customer can filter the products not only in the lowest level category but also in any higher-level category.
- Customers can display all the products of a trader.

Use the following five data structures properly (do NOT use simple array structure) to implement functionalities of the application efficiently:

- ArrayList
- LinkedList
- Queue
- Tree
- Hash Table

You need to use each of them at least once, but you feel free to use more than once when you needed. Indicate which data structure is used to implement which part of the application and why you do it so in the report file. Use different sorting algorithms (your implementation) while sorting the search results by the price, the discount percentage, or the name.

There is a CSV file on the Moodle page which is named as “e-commerce-samples.csv”. Use the given csv file to create your data files. Keep the users, the orders, and the products data in your own text files. When a customer performs a search, you need to access to the products file, and return the suitable products as the result of the search. **Don’t load all the products into the memory.** You need to keep products in a well-organized data file for products to answer queries efficiently.

### RESTRICTIONS:

- Can be only one main class in project
- Don't use any other third part library

### GENERAL RULES:

- For any question firstly use **course news forum** in Moodle, and then the contact TA.
- You can submit assignment one day late and will be evaluated over sixty percent (%60).

### TECHNICAL RULES:

- You must write a driver function that demonstrates all possible actions in your homework. For example, if you are asked to implement an array list and perform an iterative search on the list then, you must at least provide the following in the driver function:
  - o Create an array list and add items to the list. Append items to head, tail, and  $k^{th}$  index of the list.
  - o Perform at least two different searches by using two items in the list and print the index of the items.
  - o Perform another search with an item that isn't in the array list and inform the user that the item doesn't exist in the array list.
  - o Delete an existing item from the list and repeat the searches.
  - o Try to delete an item that is not on the array list and throw an exception for this situation.

The driver function should run when the code file is executed.

- Implement [clean code standards](#) in your code;
  - o Classes, methods and variables names must be meaningful and related with the functionality.
  - o Your functions and classes must be simple, general, reusable and focus on one topic.
  - o Use standard [java code name conventions](#).

### REPORT RULES:

- Add all [javadoc](#) documentations for classes, methods, variables ...etc. All explanation must be meaningful and understandable.
- You should submit your homework code, Javadoc and report to Moodle in a "studentid\_hw6.tar.gz" file.
- Use the given homework format including **selected parts from the table below**:

Detailed system requirements	
The Project use case diagrams (extra points)	
Class diagrams	X
Other diagrams	

Problem solutions approach	X
Test cases	X
Running command and results	X

#### GRADING :

- **No OOP design:** -100
- **No interface:** -95
- **No method overriding:** -95
- **No error handling:** -50
- **No inheritance:** -95
- **No polymorphism:** -95
- No javadoc documentation: -50
- No report: -90
- Disobey restrictions: -100
- **Cheating:** -200
- Your solution is evaluated over 100 as your performance.

#### CONTACT:

- Teaching Assistant: Mehmet Burak Koca
- b.koca@gtu.edu.tr