Military Institute of Science and Technology
B.Sc. in Computer Science and Engineering
Online (Fall-2022)

Subject: CSE-220, Object Oriented Programming II

Time: 1 hour 30 minutes **Full Marks: 40**

ID:	NAME:

Suppose, **Mr. A** is a local businessman. He started a vendor business quite a long time ago. But his business was not profitable enough. So, he decided that he needs to catch the online market as almost everyone has access to a smartphone and the internet nowadays. So, he reached out to you as he got to know that you are studying **CSE** and can develop any system. The following requirements are given by him:
He wants a system where there will be menu option as follows:

1. Customers can see the list of items and prices. The item names and prices will be stored **in a file**.
2. Customers can choose any item to buy and **put that item in a cart**.
3. Customers can **search for any item** from the cart and then **remove it from the cart**.
4. Customers can **see the final/total price** of the items added to the cart.
5. Customers can **see the bill/brochure** containing
   the item name - quantity of item - total price of that item.

Now, dear future aspiring CSE graduate, help Mr. A to build the system, containing the following classes and functions. There are 3 classes to implement the functionalities given above:

Class	**Attributes**	**Methods**
Product	private String name; private int price; private int quantity;	● Constructors ● String getName(); ● int getPrice(); ● int getQuantity();
Cart	List<**Product**> item;	● Constructors ● void print_menu(); ● void putIteminCart(); ● void removeItemfromCart(String item_name); ● int total_Bill(); ● void viewBrochure(); **Note:** You may use built-in functions available in java for List.
Shopping	-	● public static void main();

The description of the user-defined Methods of the **"Product" and "Cart"** class is illustrated in the following table:

Method	**Description**
String getName();	Return the ***name*** of the ***Product***.
int getPrice();	Return the ***price*** of the ***Product.***
int getQuantity();	Return the ***quantity*** of the ***Product.***
void print_menu();	Show the list of items and prices, which are stored in a file. Print it in this format: *name: xxx, price: xx taka.* **for example,** *name: Pen, price: 10 taka*
void putIteminCart();	Add a product to the list. ● Take product ***name*** and product ***quantity*** as user input. ● Find the corresponding product price from the given file. ● Create a ***Product object*** using the name, quantity, and price and add the object to the Item list and print ***"Item xxx added to the Cart!"***
void removeItemfromCart(String item_name);	Remove the ***product object*** from the list. ● If the product is in the Cart, then print ***"Item xxx removed from the Cart"*** ● Otherwise, print ***"Item xxx not found in the Cart"***

int total_Bill();	• Calculate the total Bill of the cart items using the following formula: **Total = ∑(Item price * Item quantity)** • If the bill is less than 100 tk then **throw an exception** with the following message **"Bill can't be less than 100 taka".** Otherwise, return the total bill.
void viewBrochure();	Show the following information about the items added to the shopping cart. *Item Name - Item Quantity - **Total** Price of the corresponding Item*

Sample Input	**Sample Output**
1 2 Salt 5 2 Egg 12 2 Rice 4 3 Salt 3 Pen 4 5 6	name: Pen, price: 10 tk name: Book, price: 50 tk name: Bread, price: 20 tk name: Salt, price: 50 tk name: Chips, price: 10 tk name: Egg, price: 100 tk name: Biscuit, price: 40 tk name: Teabag, price: 70 tk name: Rice, price: 60 tk Item Salt added to the Cart! Item Egg added to the Cart! Item Rice added to the Cart! Item Salt removed from the Cart Item Pen not found in the Cart Total Bill: 1440 tk Name: Egg - Quantity: 12 - price: 1200 tk Name: Rice - Quantity: 4 - price: 240 tk Exited from the Cart...

Write your code on top of the given template. Follow the given output format and function signatures closely.		
Marking Criteria	**Allocated Marks**	**Obtained Marks**
Design the classes in Java	10	
Extracting information from the File	5	
Implementation of the functions	20	
Handling the user-defined Exception	5	
Total	**40**	

The list of BuiltIn methods for List in Java is given below for your reference.

- `add()` - adds an element to a list
- `addAll()` - adds all elements of one list to another
- `get()` - helps to randomly access elements from lists
- `iterator()` - returns iterator object that can be used to sequentially access elements of lists
- `set()` - changes elements of lists
- `remove()` - removes an element from the list
- `removeAll()` - removes all the elements from the list
- `clear()` - removes all the elements from the list (more efficient than `removeAll()`)
- `size()` - returns the length of lists