

## **CENG 112 - Data Structures**

### **Assignment 1: The Shopping App**

This assignment covers the topics of:

- Strings
- Arrays
- File I/O
- ADTs
- Generics
- Bags

You are expected to implement “The Shopping App” using Java. In the app, the user has a fridge to fill and she/he can go to the mall to buy groceries.

When the user goes to the mall to shop, she/he has a basket with a size of 2000 grams. The user adds products to the basket. During shopping, the user can check the items that are in the basket already. If an item that the user selects cannot be added to the basket because of exceeding the basket capacity, your program should print a warning message. Shopping is finished when the user enters finish shopping, or the basket is full.

The items in the inventory of the mall are listed together in “*inventory.txt*” file where each line is formed as:

**`item_name,item_compartment,item_weight(grams)`**

After finishing shopping, the user heads home to fill her/his fridge. The fridge has four compartment with their limited size:

<b>[0]</b>	<b>Vegetables and Fruits</b>	3000	grams
<b>[1]</b>	<b>Meats</b>	5000	grams
<b>[2]</b>	<b>Beverages</b>	4000	grams
<b>[3]</b>	<b>Snacks</b>	2000	grams

Each item has to be put into their corresponding compartment. During the program run, you should be able to check the status of the fridge (remaining capacities of each compartment). If some of the items cannot be put to their corresponding compartments according to their remaining capacity, your program has to print a warning message. (e.g. If the remaining capacity of Vegetables and Fruits is 200 grams and you try to add a tomato to the fridge, the program should print a message like “tomato cannot be added to the fridge”).

After filling the fridge, the user can go shopping again, and return back to fill her/his fridge. Your program should end when the user enters exit, or all compartments are full.

Your code must have the interface, class and method implementations given below. The bullets (I), (C), and (M) stand for interface, class, method respectively.

#### I. IBag

- M. public boolean add(T newItem);
- M. public boolean isEmpty();
- M. public boolean isFull();
- M. public T removeByIndex(int index);
- M. public T remove();
- M. public T remove(T item);
- M. public int getItemCount();
- M. public int getIndexOf(T item);
- M. public boolean contains(T item);
- M. public void displayItems();
- M. public void dump(); // removes all the items from the bag
- M. public boolean transferTo(IBag<T> targetBag, T item);

#### C. ShoppingApp

- M. public static void main(String[] args);

#### C. FileIO

- M. public static InventoryBag<Item> readInventory();

#### C. Item

- M. public String toString();
- M. public boolean equals(Object obj);

#### C. ShoppingBasket implements IBag

#### C. MeatsCompartment implements IBag

#### C. VegetablesFruitsCompartment implements IBag

#### C. BeveragesCompartment implements IBag

#### C. SnacksCompartment implements IBag

p.s. You can add more classes and methods to your program according to your design.

NOTE: While implementing your program please make sure that your program is user friendly. Try to make your user inputs simpler. For example, in a selection process rather than asking the user to write or type a long string, make the selections with numbers.

## **Assignment Rules**

- This is a group assignment (2 students). However, inter-group collaboration is not allowed!
- All assignments are subject to plagiarism detection and the suspected violations (the solutions derived from or inspired by the solution of other groups) cause to be graded as zero.
- It is not allowed to use Java Collections Framework.
- Your code should be easy to read and test:
  - Keep your code clean. Avoid duplication and redundancy.
  - Follow Java Naming Conventions.
  - Use relative paths instead of absolute ones.

## **Submission Rules**

All submissions must:

- be performed on Microsoft Teams by only one of the group members,
- be performed before the deadline,
- be exported as an Eclipse Project and saved in ZIP format,
- include all necessary data files (txt, csv, json, etc.) in the right directory,
- follow a specific naming convention such that CENG112\_HW1\_*groupID*.

**Eclipse Project:** CENG112\_HW1\_*G5*

**Exported Archive File:** CENG112\_*G5*.zip

Submissions that do not comply with the rules above are penalized.