TCP1201 Objected-Oriented Programming and Data Structures

Assignment 1 (20%)

Trimester 2, Session 2021/2022
Faculty of Computing and Informatics
Multimedia University

DUE DATE: 27 March 2020 (Sunday), 11:59pm

1. GROUPING

1) This is a group assignment with **3-4** students per group. You may form a group if all members are under the same tutor for labs.
    - TT3V and TT5V are under Dr. Wong.
    - TT1V, TT2V, TT3L and TT6L are under Mr. Neoh.
2) The place to register your group is at MMLS (under Students > Groups).
3) Start forming group and do the assignment as early as possible. To detect sleeping member early, the group shall have meeting at least once a week, and every member shall present his/her work to the group. This is particularly important if you have a new groupmate that has never worked with. A group may split from sleeping or non- contributing members.
4) No late work will be accepted, and no deadline will be extended.
5) Do not share your code with any other group. If detected all parties involved will get zero marks.

2. TASKS

You are developing an aid distribution system. The aid can be any item. The main 3 roles in the system are listed below:

1) Donor – A donor is an individual or organization who donates aid to a distribution center (DC). A donor has a name and a phone. A donor can perform the following tasks in the system:
    a) Register and login an account at a DC.
    b) Enter the aids (name, quantity, etc.) to be donated.
    c) View the list of aids donated and the NGOs receiving the aids (tabular format).

2) NGO – An NGO (non-governmental organization) is an organization that receives aids from a distribution center. An NGO has a name and manpower count. An NGO can perform the following tasks in the system:
    a) Register and login an account at a DC.
    b) Enter the aids (name, quantity, etc.) needed.
    c) View the list of aids received and the donors of the aids (tabular format).

3) DC – A DC (distribution center) is the place where the matching of aids is performed. A DC keeps a list of donors and NGOs registered under it. A DC can perform the following tasks in the system:
    a) View all aids donated, its donors and the NGOs (tabular format).
    b) Match aids 1-to-1 (1 donor to 1 NGO)

Example: Donor D1 donates 3 blankets, NGO N1 needs 3 blankets. The result is as follows:

```
Donor Phone          Aids       Quantity   NGO   Manpower
D1    012-1111111    Blanket    3          N1    10
```

c) Match aids 1-to-many (1 donor to many NGOs)
   Example: Donor D1 donates 5 boxes of water, NGOs N1 and N2 need 3 and 2 boxes respectively. The result is as follows:

```
Donor Phone          Aids       Quantity   NGO   Manpower
D1    012-1111111    Water      3          N1    10
D1    012-1111111    Water      2          N2    20
```

d) Match aids many-to-1 (many donors to 1 NGO)
   Example: Donors D1 and D2 donate 3 and 2 boxes of biscuits respectively, NGO N1 needs 5 boxes. The result is as follows:

```
Donor Phone          Aids       Quantity   NGO   Manpower
D1    012-1111111    Biscuit    3          N1    10
D2    012-1111112    Biscuit    2          N1    10
```

e) Match aids many-to-many (many donors to many NGOs)
   Example:
   Donors D1, D2, and D3 donate 10, 20 and 30 bags of rice respectively, NGOs N1 and N2 need 15 and 40 blankets respectively. The result is as follows:

```
Donor Phone          Aids       Quantity   NGO   Manpower
D1    012-1111111    Rice       10         N1    10
D2    012-1111111    Rice       5          N1    10
D2    012-1111112    Rice       15         N2    20
D3    012-1111113    Rice       25         N2    20
D3    012-1111113    Rice       5          -     -
```

Design your classes, data fields, and methods wisely. You may add classes and data fields to support inheritance, association, aggregation, composition, or other features. Do not over design (adding classes, data fields, or methods that do not help solving the assignment, e.g., full address).

To make testing easier and save time during interview, your program should never clear screen if your program is a console application.

3. RECOMMENDED TASK DISTRIBUTION

To maximum separation of work:

- Every group member handles a specific role. For a group of 4 members, the 2 weakest members may co-develop a role.
- Develop each role separately and store the data in csv/json files. The format of the csv/json files should be agreed by members so that it can be read/write from programs developed by different members.

4. SUBMISSION

- Submit your assignment to your tutor. Check with your tutor on the submission channel.
- One group submit one zip file named *GroupID.zip* where:

GroupID – your registered group number

The zip should contain the following structures:

i. A code folder which stores all Java code files and data files. Make sure the code can be compiled and executed.
ii. A html folder which stores the Java documentation for Donor, NGO, and DC classes.
iii. A file named Members.txt that lists down the group members' id, name, and the role he/she developed.

```
TASK DISTRIBUTION
Member1 Id Name - Role
Member2 Id Name - Role
Member3 Id Name - Role
Member4 Id Name - Role
```

Mark Sheet (20%)

Criteria	Items (Mark for an item is awarded if it works and student can explain)
1. Program Execution (12 marks)	**1.1. Correct program features and output (10m)** The output must be adequate. Donor a) Register and login an account. (1m) b) Enter the aids to be donated (name, quantity, etc.) (0.5m) c) View all aids donated and the NGOs receiving the aids (tabular format). (1m) NGO a) Register and login an account. (1m) b) Enter the aids needed (name, quantity, etc.) (0.5m) c) View all aids received and the donors of the aids (tabular format). (1m) Distribution Center (DC) a) View all aids donated to the DC, the donors of the aids and the NGOs receiving the aids (tabular format) (1m). b) Match aids 1-to-1 (1m) c) Match aids 1-to-many (1m) d) Match aids many-to-1 (1m) e) Match aids many-to-many (1m)
	1.2. User friendliness (2m) JavaFX 1.0m – At least one role is fully developed in JavaFX. 0.5m – At least one role is partially developed in JavaFX. Input and Output (I/O) 1.0m – I/O are clear. Input errors are handled. 0.5m – I/O are ambiguous, or input errors are not handled. 0.0m – The program is unusable.
2. OOP Design and Java Documentation (6 marks)	**2.1. Association, Aggregation, and/or Composition (2m)** Only your own classes are counted. Using built-in classes such as String are not counted. 2.0m – Excellent (sensible modeling with good identifiers) 1.5m – Good (sensible modeling with poor identifiers) 1.0m – Average (the modeling has some issues) 0m – Do not use
	2.2. Inheritance (2m) 2.0m – Excellent (sensible modeling with good identifier names) 1.5m – Good (sensible modeling with poor identifiers) 1.0m – Average (the modeling has some issues) 0.0m – Do not use
	2.3. Java Documentation for Donor, NGO, and DC (2m) Descriptions for class, methods, parameters, and returns must be adequate. 2.0m – All 4 types of descriptions are fine 1.5-0.5m – Issues with 1-3 types of descriptions (minus 0.5m per description). 0.0m – No Java Documentation
3. Bonus (2 marks)	1m for each of the following items: i. Use files to save and share data.

	ii. Additional role is fully developed in JavaFX. iii. A role is developed in Android. iv. Any other significant feature agreeable by your tutor, e.g., able to match by date.
4. Presentation & Interview (2 marks)	2.0m – Very well presented and Q&A 1.5m – Overall fine with minor issues 1.0m – Average 0.5m – Poorly prepared or has major issues
5. Sleeping member, late submission, not attending interview, or plagiarism	0 mark for this assignment