SIM GLOBAL EDUCATION

UOW AUSTRALIA

SCIT

School of Computing and Information Technology
Faculty of Engineering & Information Sciences

CSIT121
Object Oriented Design and Programming
Assignment 2
Due date: 8 May 2022

Objectives

- Apply OO design technique: allocation of classes and responsibilities.
- Use OO features: class, inheritance and polymorphism to implement the solution.
- Adopt good programming practices: modular code, appropriate comments, etc.

Programming Task (8 marks)

Write a Java program to assist the management of a toy car manufacturer to manage its inventory. There are two categories of toy cars: ordinary toy car and electric toy car. The details are as follow.

Ordinary toy car	Self pedaling toy car
Information required	Model code • A unique code for each model. Price • The selling price of this model. Quantity • The number of units currently available in the warehouse.
Calculation required	Total inventory worth • Price x Quantity Insurance cost • 2% of the total inventory worth.

Electric toy car	Battery-powered toy car
Information required	Model code • A unique code for each model. Price • The selling price of this model. Quantity • The number of units currently available in the warehouse. Battery duration • How long (in minutes) the battery can last upon fully charged. Charging duration • How long (in minutes) it takes to charge the battery.
Calculation required	Total inventory worth • Price x Quantity Insurance cost • 10% of the total inventory worth

Define two classes: ToyCar (super class) and ToyCarElect (subclass) to represent the above. You must choose an appropriate data type for each variable. You may include additional variables as you deemed fit. You must also include appropriate get/set methods for each class in addition to the methods specified above.

The program (a mini-inventory management software) will allow the user (manager) to perform the following operations repeatedly until he decides to quit.

```
1 Add inventory
2 Remove inventory
3 Show all inventory
4 Search inventory by car model
5 Search inventory by car price
6 Search inventory by car battery duration
7 Quit
```

Option 1 Add inventory

The program will allow the user to enter the information of a ToyCar or ToyCarElec, check that the model is not already in the inventory and add to the inventory (ArrayList).
If the model is already in the inventory, the program will display an appropriate message and will not add the information to the inventory.

The program must validate the input values when appropriate (e.g. price > 0, quantity >= 0, etc.). You will decide where (in the program) and how to perform the validations, and how to respond accordingly.

Option 2 Remove inventory

The program will prompt the user to enter a model code, find the model (instance) in the inventory (ArrayList) and remove it. If the model does not exist, the program must display an appropriate message. If the model is found but the quantity is not zero, the program must not remove the instance. Instead, the program will inform the user accordingly.

Option 3 Show all inventory

The program will display all models (instances) in the inventory. You must decide (reasonably) what to display for each model (instance).

Option 4 Search inventory by car model

The program will prompt the user for a model, search and display the matching instance (appropriate information).

Option 5 Search inventory by car price

The program will prompt the user to enter two values: the lower bound and upper bound. The program will display all models whose price is within the range (inclusive at both ends). If there is no matching instance, the program will display an appropriate message.

Option 6 Search inventory by battery duration

The program will prompt the user to enter a value (battery duration in minutes), and display models whose battery duration >= the input value. If there is no matching instance, the program will display an appropriate message.

Option 7

The program will exit the loop and terminate.

Source code comments

Please include appropriate header and block comments in the program.

Submission

Please submit a single Java file (containing the above classes) to Moodle.