Assigment 02 – Inheritance

Duration: 70'

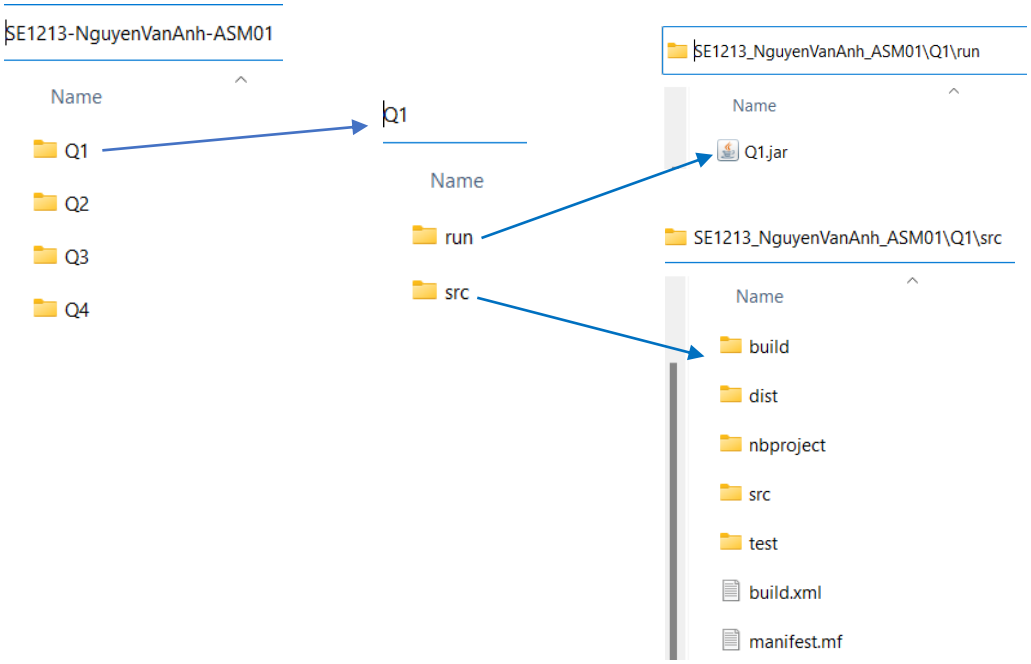
==========

Software Requirements

- Netbean 8.2 or later, JDK 8 or later, Notepad, Command Prompt, WinRAR / WinZip with Windows Explorer (File Explorer) on Windows 7 and above.

Instructions

- Step 1: Students download the given materials from LMS.
- Step 2: Students read questions and prepare answers in the given template.
- Step 3: Prepare to submit the answer:
  - For each question (e.g., question **Q1, Q2, Q3,…**), please create two sub-folders: **run** and **src**.
  - Copy the *.**jar** file into the **run** folder, and the entire project source code file into the **src** folder.
- Step 4: Submit a solution for each question:
  - Create a folder named : RollNumber_FullName_ASM0x (**x : 1, 2, or 3**) that contains folders (created Step 03) as the below figure:



  - Use WinRAR / WinZip tool to compress the **RollNumber_FullName_ASM0x** folder and submit it to LMS

❖ **Importance:**

- *Do not change the names of the folders specified (or required) in the exam and the name of submit folder in Step 04 must be correct. If you change it (incorrectly), the grading software can not find the execute file (.jar) to score, thus the exam result will be 0.*

==========

❖ **Question 2: (10 marks)**

Write a class **Vehicle** and a class **Audi** extending from **Vehicle** (i.e. Vehicle is a superclass and Audi is a subclass) with the following information:

Vehicle
-id: String -name: String -price: double

Where:

- Check validation (apply to constructors and setters):

-quantity: int
+Vehicle() +Vehicle(id:String, name:String, price:double, quantity:int) +getId():String +setId(value:String):void +getName (): String +setName(value:String):void +setPrice(value:double):void +getPrice (): double +setQuantity(value:int):void +getQuantity (): int +getSubTotal():double +override toString():String

- check the **id** is not empty and format : Vxxxxxx (x is digits). If value is invalid then set id to V000000
- check the **name** is not empty and length from 5 to 50 characters. If value is invalid then set name to "new vehicle".
- check the **price** from 1 to 1.000.000. If value invalid then set price to 1
- check the **quantity** from 1 to 1.000. If value invalid then set quantity to 1

- Vehicle() - default constructor (numeric value is 0, string value is empty)
- Vehicle(id:String, name:String, price:double, quantity:int): constructor, which sets values to id, name, price, quantity)
- getName ():String – return name with title case and each words is seperated by a space
- getSubTotal():double– return price * quantity.
- toString():String – return the string of format :**id, name, price, quantity, subTotal** (the **price** and **subTotal** are formatted with two decimal places).

Where:

Audi
-releasedYear: int -color: String
+Audi() +Audi(releasedYear: int, color: String) +override toString():String +override getSubTotal (): double

- Check validation (apply to constructors and setters):
  - check the **releasedYear** from 2022 to 2025. If the value is invalid then set releasedYear to 2022
  - check the **color** in [Black, White, Red, Gray]. If the value is invalid then set the color to "Black"
- Audi() - default constructor (numeric value is 0, string value is empty)
- Audi(id: String, name: String, price: double, quantity: int, color: String, releasedYear:int): constructor, which sets values to id, name, price, quantity, color, releasedYear
- toString():String – return the string of format : **id, name, color, releasedYear, price, quantity, subTotal** (the **price** and **subTotal** are formatted with two decimal places).
- override getSubTotal (): double – sub total = price * quantity * x% (default x = 100) . If color is red or releasedYear is 2025 then x = 120

Do not format the result.

The program output might look something like this:

```
#Case 1:  [2.0 marks]
1.Test Vehicle : toString()
2.Test Audi :toString()
Enter Test Case No.(1 | 2 ):1
Enter id:v010203
Enter name:audi v8
Enter price:5000
Enter quantity:2
OUTPUT:
V010203,Audi V8,5000.00,2,10000.00
--------------------------------------------------

#Case 2:  [2.0 marks]
1.Test Vehicle : toString()
2.Test Audi :toString()
Enter Test Case No.(1 | 2 ):1
Enter id:v020304
Enter name:audi v8
Enter price:5000000
Enter quantity:2000
OUTPUT:
V020304,Audi V8,1.00,1,1.00
--------------------------------------------------

#Case 3:  [2.0 marks]
1.Test Vehicle : toString()
2.Test Audi :toString()
Enter Test Case No.(1 | 2):2
Enter id:v0908072
Enter name:audi v8
Enter price:900
Enter quantity:2
Enter color:Red
Enter release year:2024
OUTPUT:
V000000,Audi V8,Red,2024,900.00,2,2160.00

--------------------------------------------------

#Case 4:  [2.0 marks]
1.Test Vehicle : toString()
2.Test Audi :toString()
Enter Test Case No.(1 | 2):2
Enter id:v09082376
Enter name:audi limited
Enter price:7000
Enter quantity:2
Enter color:green
Enter release year:2025
OUTPUT:
V000000,Audi Limited,Black,2025,7000.00,2,16800.00
--------------------------------------------------

#Case 5:  [2.0 marks]
1.Test Vehicle : toString()
2.Test Audi :toString()
Enter Test Case No.(1 | 2):2
Enter id:v001
Enter name:abc
```

```
Enter price:0
Enter quantity:0
Enter color:abc
Enter release year:1999
OUTPUT:
V000000,New Vehicle,Black,2022,1.00,1,1.00
```