CDAC Mumbai
PG-DAC Mar22 Batch (Juhu & Kharghar)

Time : 2hrs

Marks : 30

--

Problem 1 : Accelerate the Car (10 Marks)

a) Create a new Java class named **Car** that has the following **private** fields
- **year** - The year field is an **int** that holds a car's year model (e.g. 2010)
- **make** - The make field is a **String** object that holds the make of the car (e.g. "Porsche")
- **speed** - The speed field is an **double** that holds a car's current speed (e.g. 25.0)

b) In addition, the Car class should have the following methods.
- **Constructor** - The constructor should accept the car's year, make, and beginning speed as arguments
  - ◆ These values should be used to initialize the Car's year, make, and speed fields
- **Getter Methods** - Write three accessor (getter) methods to get the values stored in an object's fields
- **accelerate** - Write an **accelerate** method that has no arguments (parameters) passed to it and adds **1** to the **speed** field each time it is called
  - ◆ For example: if the car was going 3 mph, accelerate would set the speed to 4 mph

c) Write a separate java class **RaceTrack** in a separate file with a **main()** method that
- Create a new **Car** object (using the Car constructor method), passing in the year, make, and speed
- Display the current status of the car object using the getter methods **getYear**(), **getMake**(), and **getSpeed**()
- Call the car's accelerate method and then re-display the car's speed using **getSpeed**()

Problem 2 : Inventory Management (20 Marks)
Write a program to create an inventory of items which will allow basic inventory management such as below
==

1) **Add Item (Prevent duplication)**
2) **Display complete inventory in sorted order of item names as well as itemId.**
3) **Remove Item.**
4) **Exit**

Please enter your choice (1-3)

1. Define a class **Item** with two attributes **itemId** and **itemName**.
2. The implementation should use **ArrayList** collection where every element of ArrayList collection holds the object of class Item.
3. Put constraint on ArrayList of Item objects that, it should not allow insertion when the values of attributes itemID and itemName previously exist **together** in the ArrayList.

Ex : If itemId and itemName with values (1, “Item1”) respectively, already exists in the **ArrayList** Collection, then same entry should not exist in your collection.