Course: Object-Oriented Programming (Spring 2023)

Resource Person: Shazma Noor Assignment–2 (Objects & Classes)

Total Points: 30

Submission Due: Monday July 24, 2023

Instructions: Please Read Carefully!

- This is an <u>individual</u> assignment. Everyone is expected to complete the given assignment on their own, without seeking any help from any website or any other individual. There will be strict penalties for any work found copied from any source and the university policy on plagiarism will be strictly enforced.
- You are expected to submit this assignment as:
 - Create a single .java file solution of the assignment question. The name of the .java file should be as mentioned in the assignment question.
 - Execute the program to test the complete working and take screenshots of cmd.
- Java File and cmd screenshots of each menu operation should be submitted in googleclassroom under Assignment thread.
 - Submit your assignment on or before due date. Late Submission will get -25% each day.

You are being asked to implement these **UML** diagrams and utilize them using the **main** function.

1. Implement **Vehicle.java** for the UML given below:

```
Vehicle
- vehicleType : String
- numberOfWheels: int
- seatingCapacity : int
- speed: double
 + setVehicle(vehicleType : String) : void
 + setNumberOfWheels (numberOfWheels: int): void
 + setSeatingCapacity(seats : int) : void
 + setSpeed(speed : double) : void
 + getVehicleType(): String
 + getNumberOfWheels(): int
 + getSeatingCapacity(): int
 + getSpeed(): double
 + Vehicle()
 +Vehicle(vehicleType:String,numberOfWheels:int,
 seatingCapacity : int, speed : double)
 +Vehicle(vehicleType:String,speed:double
 ,numberOfWheels:int, seatingCapacity:
 int)
 + deepCopy(vehicle : Vehicle) : void
 + shallowCopy(vehicle: Vehicle): void
 + getCallerObject(): Vehicle
 + toString(): String
```

Here, the **getCallerObject()** shall return that object through which the call to this method was made. Or Simply put, it shall return that particular object who called this method.

2. Create a **Car.java** for given UML:

Car - modelName : String - licenseNumber : String - yearOfPurchase: int + Car() +Car(modelName:String,licenseNumber:String, yearOfPurchase : int) +Car(modelName:String,licenseNumber:String) + setModelName(modelNmae : String) : void + setLicenseNumber (numberOfWheels : String) : void + setYearOfPurchase(seats : int) : void + getModelName(): String + getLicenseNumber (): String + getYearOfPurchase(): int + Copy(Car car): void + toString(): String

3. Implement Engine.java using this UML:

Engine - capacity: float - numberOfCylinders: int + Engine() + Engine(capacity:float, numberOfCylinders:int) + setCapacity(capacity: float): void + setNumberOfCylinders(numberOfWheels:String):void + getCapacity(): float + getNumberOfCylinders(): int + equals(Engine engine): boolean + toString(): String

You have to **override** and thus implement the **equals**() method from **Object** class such that it shall return **true** when both the capacity and numberOfCylinders of caller and argument object are equal to each otherand **false** otherwise. (Try to do this in one line.)

4. Finally, create a **Main** class having main function. Create instances / objects of these classes anduse all of the methods/functions. Also display the information of all these instances using **toString()** which is implemented.