**Java 236 – Assignment 5**

**Vehicle Management System**

Develop a vehicle management system that showcases the concepts of inheritance and polymorphism.

Requirements:

1. Create a base class called **Vehicle**, which should have the following attributes:
   * **make** (String): The make of the vehicle (e.g., Toyota, Ford).
   * **model** (String): The model of the vehicle (e.g., Camry, Mustang).
   * **year** (int): The manufacturing year of the vehicle.
   * **price** (double): The price of the vehicle.
2. Implement subclasses for **Car**, **Truck and SUV**. Each subclass should inherit from the **Vehicle** class and include additional attributes specific to cars, trucks and SUVs respectively. For example: (this is not an inclusive list)
   * **Car** should have attributes like **numDoors** (int) and **isConvertible** (boolean).
   * **Truck** should have attributes like **bedLength** (double) and **towingCapacity** (double).
   * **SUV should have seating capacity and third row availability.**
3. Implement appropriate constructors and methods in each class to set and retrieve the values of attributes.
4. Create a **Main** class to demonstrate the use of inheritance and polymorphism. In the **Main** class:
   * Implement a method to allow users to create cars by providing input for attributes such as make, model, year, price, number of doors, and convertible status.
   * Save the created cars to an array.
   * Utilize polymorphism to treat instances of Car and Truck as instances of the base class Vehicle.
   * Display the details of each vehicle, including both inherited and subclass-specific attributes.
   * Add any other options to allow the user to have a full application experience
5. Ensure that the program demonstrates a clear understanding of inheritance and polymorphism concepts, with appropriate method overriding or overloading where necessary.
6. Provide a user-friendly command-line interface or a Jpane interface with clear instructions and feedback.
7. Make sure to provide comments to your code to be consise and notate your method of implementation.
8. Save, Zip and Submit your project as yourName\_VehicleManagement

Bonus:

* Implement additional subclasses (e.g., **Boat**, **Motorcycle**) to expand the types of vehicles managed by the system.
* Add methods to calculate vehicle-related metrics, such as fuel efficiency or depreciation over time.