Q1. Create a vehicle class with an init method having instance variables as name_of_vehicle, max_speed and average_of_vehicle.

Q2. Create a child class car from the vehicle class created in Que 1, which will inherit the vehicle class. Create a method named seating_capacity which takes capacity as an argument and returns the name of the vehicle and its seating capacity.

Q3. What is multiple inheritance? Write a python code to demonstrate multiple inheritance.

Q4. Assignment: Python Object-Oriented Programming Concepts

Objective: To create Python classes implementing the concepts of polymorphism, encapsulation, constructors, and inheritance.

Task 1: Create a Python Class

Create a class called Person with the following:

1. Properties: name, age, gender
2. A constructor to initialize these properties.
3. Methods: say_hello(), which prints out a greeting from the person; and is_adult(), which returns True if the person is 18 or above, and False otherwise.

Task 2: Implement Inheritance

Create a subclass called Student that inherits from the Person class. The Student class should have the following additional properties:

1. student_id
2. course

Create appropriate methods as you deem necessary.

Task 3: Implement Polymorphism

Create another subclass called Teacher that also inherits from the Person class. The Teacher class should have the following additional properties:

1. teacher_id
2. subject

Ensure that the say_hello() method for the Teacher class is different from the one for the Person and Student classes. This will demonstrate polymorphism.

Task 4: Encapsulation

Make the age property in the Person class private and create a method to allow access to it. This is an example of encapsulation, as it restricts direct access to the age property.

Deliverable: Create your assignment in python notebook and upload it to GitHub & share that GitHub repository link through your dashboard. Make sure the repository is public.

Remember, the primary goal of this assignment is to understand the concepts of Object-Oriented Programming. It's important to comment your code to explain your thought processes and approach. Also, be sure to test your classes and methods with a variety of inputs to ensure they function as expected. Happy coding!