**Java Project Assignment: Demonstrating Inheritance**

**Project Overview:** In this assignment, you'll create a simple Java application to demonstrate the importance of inheritance in object-oriented programming (OOP). You'll design a system that includes a base class Vehicle and its derived classes Car and Truck. The goal is to show how inheritance helps in code reuse, polymorphism, and extending functionality.

**Requirements:**

1. **Base Class Vehicle**
   * Fields:
     + String make: The make of the vehicle (e.g., Toyota, Ford).
     + String model: The model of the vehicle (e.g., Corolla, F-150).
     + int year: The year the vehicle was manufactured.
   * Methods:
     + void start(): Prints "Vehicle is starting".
     + void stop(): Prints "Vehicle is stopping".
     + String getDetails(): Returns the make, model, and year as a string.
2. **Derived Class Car** (inherits from Vehicle)
   * Additional Field:
     + int doors: The number of doors in the car.
   * Methods:
     + Override start(): Prints "Car is starting".
     + Override stop(): Prints "Car is stopping".
     + Override getDetails(): Returns details of the car including the number of doors.
3. **Derived Class Truck** (inherits from Vehicle)
   * Additional Field:
     + double payloadCapacity: The payload capacity of the truck in tons.
   * Methods:
     + Override start(): Prints "Truck is starting".
     + Override stop(): Prints "Truck is stopping".
     + Override getDetails(): Returns details of the truck including payload capacity.
4. **Test Case Class**:
   * Create unit tests using **JUnit** to test your classes. Ensure that:
     + The start() and stop() methods behave as expected.
     + The getDetails() method returns the correct information.
     + The polymorphic behavior (using Vehicle reference to hold objects of Car and Truck) works correctly.

**Solution Structure:**

1. **Vehicle.java**: Base class
2. **Car.java**: Derived class
3. **Truck.java**: Derived class
4. **VehicleTest.java**: JUnit test class