1. Car and Engine Relationship (Composition)

Steps to Implement

1. **Create an `Engine` class**
   - Add properties like `horsepower` and `fuelType`.
   - Create a constructor to initialize these values.
   - Create a method `startEngine()` to print a message.
2. **Create a `Car` class**
   - Define an instance variable `engine` of type `Engine`.
   - Initialize `Engine` inside the `Car` constructor (composition).
   - Create a method `startCar()` that calls `engine.startEngine()`.
3. **Create a `main` method**
   - Instantiate an `Engine` object.
   - Instantiate a `Car` object and pass the `Engine`.
   - Call `startCar()` to see the result.

---

2. Library and Book Relationship (Aggregation)

Steps to Implement

1. **Create a `Book` class**
   - Define properties like `title` and `author`.
   - Create a constructor to initialize these values.
   - Create a method `displayBookInfo()` to print details.
2. **Create a `Library` class**
   - Define an instance variable `book` of type `Book`.
   - Create a constructor that accepts a `Book` object.
   - Create a method `displayLibraryInfo()` that calls `book.displayBookInfo()`.
3. **Create a `main` method**
   - Instantiate a `Book` object.
   - Instantiate a `Library` object with a `Book`.
   - Call `displayLibraryInfo()`.

---

3. Employee and Address Relationship (Composition)

Steps to Implement

1. **Create an `Address` class**
   - Define properties like `city` and `state`.
   - Create a constructor to initialize them.
   - Create a method `displayAddress()` to print details.
2. **Create an `Employee` class**
   - Define an instance variable `address` of type `Address`.
   - Initialize `Address` within the `Employee` constructor.
   - Create a method `displayEmployeeInfo()` that calls `address.displayAddress()`.
3. **Create a `main` method**
   - Instantiate an `Address` object.
   - Instantiate an `Employee` object with `Address`.
   - Call `displayEmployeeInfo()`.

---

4. Bank and Account Relationship (Aggregation)

Steps to Implement

1. **Create an `Account` class**
   - Define properties like `accountNumber` and `balance`.
   - Create a constructor to initialize them.
   - Create a method `displayAccountInfo()`.
2. **Create a `Bank` class**
   - Define an instance variable `account` of type `Account`.
   - Create a constructor that accepts an `Account` object.
   - Create a method `displayBankInfo()` that calls `account.displayAccountInfo()`.
3. **Create a `main` method**
   - Instantiate an `Account` object.
   - Instantiate a `Bank` object with an `Account`.
   - Call `displayBankInfo()`.