Inheritance and Polymorphism:

a. Create a class Shape with methods to calculate the area. Derive classes Circle, Rectangle, and Triangle from Shape class. Demonstrate polymorphism by creating an array of Shape objects and calculating their areas.

b. Implement a class hierarchy for a library system. Create classes LibraryItem, Book, and DVD where Book and DVD inherit from LibraryItem. Each class should have properties like title, author/director, and a unique identifier. Implement a method to display information about each item.

Encapsulation:

a. Create a class Student with private attributes like name, age, and grade. Provide public methods to set and get these attributes. Implement a constructor to initialize the student object.

b. Build a simple ATM system. Create a class ATM with methods for withdrawing, depositing, and checking balance. Implement encapsulation to protect sensitive information.

Abstraction:

a. Design an abstract class Vehicle with attributes like make, model, and methods like start(), stop(). Create concrete classes Car and Motorcycle that extend Vehicle. Implement the methods accordingly.

b. Create an abstract class Shape with abstract methods calculateArea() and calculatePerimeter(). Implement subclasses like Circle, Rectangle, and Triangle with appropriate methods.

Polymorphism:

a. Implement a class Animal with methods makeSound() and move(). Create subclasses like Dog, Cat, and Bird. Override the methods in each subclass to exhibit different behaviors.

b. Develop a simple game using the concept of polymorphism. Create a base class GameCharacter with methods attack() and defend(). Derive classes like Knight, Wizard, and Archer that override these methods with unique behaviors.

Interface and Inheritance:

a. Design an interface Drawable with a method draw(). Implement classes Circle, Rectangle, and Triangle that implement this interface. Create an array of Drawable objects and demonstrate their drawing capabilities.

b. Create an interface Payment with methods processPayment() and refund(). Implement classes CreditCardPayment, PayPalPayment, and CashPayment that implement the Payment interface.