Assignment 5

Polymorphism

1. Create a system to showcase polymorphic behavior in a payroll management application. Design a base class Employee with common attributes such as eid (employee ID), ename (employee name), and salary. Implement essential methods, including a constructor, setter, getter, display, and a member function to calculate the salary based on the specific employee type.

   Derive three classes from the base class: HR, SalesManager, and Admin. Each derived class should have its own set of attributes and methods, along with inheriting the attributes and methods from the base class.

   Employee Class:

   Attributes:

   eid: Employee ID
   ename: Employee Name
   salary: Basic Salary
   Methods:

   Constructor: Initializes the attributes.
   Setter: Sets values for attributes.
   Getter: Retrieves values of attributes.
   Display: Displays employee information.
   CalculateSalary: Calculates and returns the salary.
   HR Class (Derived from Employee):

   Additional Attributes:

   commission: Commission percentage
   Additional Methods:

   Constructor: Initializes the attributes.
   Setter: Sets values for attributes.
   Getter: Retrieves values of attributes.
   Display: Displays HR-specific information.
   CalculateSalary: Overrides the base class method to calculate HR's total salary including commission.
   SalesManager Class (Derived from Employee):

   Additional Attributes:

   target: Sales target
   incentive: Incentive amount
   Additional Methods:

Constructor: Initializes the attributes.
Setter: Sets values for attributes.
Getter: Retrieves values of attributes.
Display: Displays Sales Manager-specific information.
CalculateSalary: Overrides the base class method to calculate Sales Manager's total salary including incentives.
Admin Class (Derived from Employee):

Additional Attributes:

allowance: Allowance amount
Additional Methods:

Constructor: Initializes the attributes.
Setter: Sets values for attributes.
Getter: Retrieves values of attributes.
Display: Displays Admin-specific information.
CalculateSalary: Overrides the base class method to calculate Admin's total salary including allowances.

2. Demonstrate polymorphic behavior in a geometric shapes application. Create a base class Shape with a common attribute area. Include essential methods such as a constructor, a member function to calculate the area, and another member function to display the shape's information.

Derive three classes from the base class Shape: Triangle, Circle, and Rectangle. Each derived class should have its own set of attributes (height and base for Triangle, radius for Circle, length and width for Rectangle) and methods. The derived classes must inherit the area attribute and methods from the base class.

Shape Class (Base Class):

Attributes:

area: Area of the shape
Methods:

Constructor: Initializes the attributes.
Setter: Sets values for attributes.
Getter: Retrieves values of attributes.
CalculateArea: Calculates and sets the area based on the shape.
Display: Displays the shape's information, including the calculated area.
Triangle Class (Derived from Shape):

Additional Attributes:

height: Height of the triangle
base: Base of the triangle
Additional Methods:

Constructor: Initializes the attributes.
Setter: Sets values for attributes.
Getter: Retrieves values of attributes.
CalculateArea: Overrides the base class method to calculate the area of the triangle.
Display: Overrides the base class method to display triangle-specific information.
Circle Class (Derived from Shape):

Additional Attributes:

radius: Radius of the circle
Additional Methods:

Constructor: Initializes the attributes.
Setter: Sets values for attributes.
Getter: Retrieves values of attributes.

CalculateArea: Overrides the base class method to calculate the area of the circle.
Display: Overrides the base class method to display circle-specific information.
Rectangle Class (Derived from Shape):

Additional Attributes:

length: Length of the rectangle
width: Width of the rectangle
Additional Methods:

Constructor: Initializes the attributes.
Setter: Sets values for attributes.
Getter: Retrieves values of attributes.

CalculateArea: Overrides the base class method to calculate the area of the rectangle.
Display: Overrides the base class method to display rectangle-specific information.

3. Write a code to show polymorphic behavior where vehicle is base class and derived classes like bike, car, bus etc. Override the break function.
4. Write 2 more codes to show polymorphic behavior on your own