

Week 5 Practice Questions - Object Oriented Programming Paradigm

I. Abstraction:

Qn 1: Create an abstract class Animal with abstract methods make Sound() and eat(). Create subclasses Dog, Cat, and Bird that inherit from Animal and implement the abstract methods accordingly.

Qn 2: Create an abstract class Shape with abstract methods calculateArea() and calculatePerimeter(). Create subclasses Circle, Rectangle, and Triangle that inherit from Shape and implement the abstract methods accordingly.

Qn 3: Create an abstract class Employee with abstract methods calculateSalary() and getEmployeeDetails(). Create subclasses RegularEmployee, ContractEmployee, and HourlyEmployee that inherit from Employee and implement the abstract methods accordingly.

II. Polymorphism:

Qn 1: Create a hierarchy of animals with classes like Animal, Mammal, Bird, and Fish. Implement polymorphism to demonstrate different animal sounds.

Hint:

Create an abstract class Animal with an abstract method makeSound().

Create subclasses Mammal, Bird, and Fish, inheriting from Animal and overriding the makeSound() method.

Create an array of Animal objects to store different types of animals.

Iterate through the array and call the makeSound() method for each animal, demonstrating polymorphism.

Qn 2: Overload

Create a Calculator class with multiple calculate () methods that take different combinations of arguments (e.g., calculate (int, int), calculate (double, double), calculate (int, int, char)). Implement these methods to perform different arithmetic operations based on the arguments.

Qn 3: Override

Create a base class Calculator with a calculate () method that takes two double arguments. Implement this method to perform addition. Create subclasses SubtractionCalculator, MultiplicationCalculator, and DivisionCalculator that inherit from Calculator and override the calculate () method to perform their respective arithmetic operations.

III. Inheritance:

Qn 1: Base Class: Animal

- Attributes: name, sound, num_legs
- Methods: make_sound(), walk()

Derived Classes:

- **Dog:** barks
- **Cat:** meows
- **Bird:** chirps, flies

Questions:

1. Create a base class Animal with the specified attributes and methods.
2. Create derived classes Dog, Cat, and Bird that inherit from Animal.
3. Override the make_sound() and walk() methods in each derived class to provide specific implementations.
4. Create objects of each derived class and demonstrate their behavior (e.g., call make_sound() and walk() methods).

Qn 2: Consider a vehicle hierarchy with a base class Vehicle that has attributes like make, model, year, and num_wheels. We want to create derived classes Car, Truck, and Motorcycle that inherit from Vehicle and have additional specific attributes and methods. Additionally, we want to introduce another base class Engine with attributes like type, horsepower, and fuel_type, and have Car, Truck, and Motorcycle inherit from both Vehicle and Engine. Implement using multiple inheritance.

[Faculty can include questions for multilevel, hybrid and hierarchical inheritance concepts if required]

IV.Encapsulation

Qn 1: A bank wants to create a class to represent bank accounts. Each account should have a unique account number, an initial balance, and methods to deposit and withdraw funds. The bank wants to ensure that the account balance is always positive and that withdrawals do not exceed the balance.

- Create a class named BankAccount with private attributes for accountNumber, balance, and a constant for the minimum balance allowed.
- Implement public getter and setter methods for the accountNumber attribute.
- Implement public methods deposit and withdraw that update the balance appropriately, ensuring that the balance remains positive and that withdrawals do not exceed the balance.

Qn 2: Write a Java program to create a class called Rectangle with private instance variables length and width. Provide public getter and setter methods to access and modify these variables.