PYTHON

Assignment

Task 1: Single Inheritance

Create a base class `Animal` with a method `sound()`. Then, create a derived class `Dog` that inherits from `Animal` and overrides the `sound()` method. Demonstrate the functionality by creating an object of the `Dog` class.

Task 2: Multiple Inheritance

Define two classes, `Mother` and `Father`, each with a method `profession()`. Create a derived class `Child` that inherits from both classes and calls the `profession()` method from both parent classes.

Task 3: Multilevel Inheritance

Create a base class `Vehicle`, a derived class `Car` that inherits from `Vehicle`, and another derived class `ElectricCar` that inherits from `Car`. Add methods in each class that represent their respective functionalities.

Task 4: Calling Parent Class Method

In a class `Person`, create a method `greet()`. In the derived class `Employee`, override the `greet()` method but also call the `greet()` method from the `Person` class. Demonstrate this by creating an object of `Employee`.

Task 5: Use of `super()` Keyword

Create a base class `Shape` with a method `__init__()` that initializes the type of shape. Create a derived class `Circle` that initializes both the type of shape and the radius. Use the `super()` function to call the parent class constructor.

Task 6: Hierarchical Inheritance

Create a base class `Plant`. Derive two classes `Tree` and `Flower` from `Plant`. Each derived class should implement its own version of a method `grow()`. Test this with objects of `Tree` and `Flower`.

7. Hybrid Inheritance Coding Challenge: University System

In this challenge, you will design a system for a university where multiple types of staff (like Professors, Administrators, and Teaching Assistants) work. The university also has different departments (e.g., Computer Science, Mathematics). Each staff member belongs to a department and has specific roles based on their job type.

Requirements:

1. **Base Class: Create a class `Person` with common attributes like `name`, `age`, and a method `get_details()` to display the person's information.**
2. **Intermediate Class: Create a class `Department` that represents a department in the university. It should contain attributes like `department_name`, and a method `get_department()` that returns the department name.**
3. **Derived Classes:**
   - **Create a class `Professor` that inherits from both `Person` and `Department`. Add an attribute `subject_specialization` and a method `teach()` that displays what the professor teaches.**
   - **Create a class `Administrator` that inherits from both `Person` and `Department`. Add an attribute `role` and a method `manage()` that describes the administrative duties.**
   - **Create a class `TeachingAssistant` that inherits from both `Person` and `Professor`. Add a method `assist()` that displays how the TA assists in the teaching process.**
4. **Hybrid Inheritance: Demonstrate hybrid inheritance by creating instances of `Professor`, `Administrator`, and `TeachingAssistant`. Each staff member should display their personal details, department, and unique role in the university system.**

Example Output:

```
Professor Name: Dr. John Smith
Age: 45
Department: Computer Science
Subject Specialization: Artificial Intelligence
Teaches: Artificial Intelligence

Administrator Name: Sarah Lee
Age: 38
Department: Mathematics
Role: Head of Admissions
Manages: Admissions Process

Teaching Assistant Name: Mike Johnson
Age: 25
Department: Computer Science
Subject Specialization: Data Structures
Assists in teaching Data Structures
```