Python OOP Practice Exercises

Inheritance

1. 1. Create base class 'Animal' and subclasses 'Dog' and 'Cat'. Implement common and unique methods.
2. 2. Develop a 'Vehicle' class and derive 'Car', 'Bike' from it. Use __init__ in both base and derived classes.
3. 3. Make a base class 'Account'. Inherit 'SavingsAccount' and 'CurrentAccount' and override a method.
4. 4. Use multilevel inheritance: Base - 'Organism', Derived - 'Animal', Further Derived - 'Human'.
5. 5. Create a 'Person' class and inherit 'Teacher' and 'Student'. Override introduction method.
6. 6. Write 'Shape' as base, and 'Rectangle', 'Circle' as subclasses. Implement area method in each.
7. 7. Create 'Appliance' class. Inherit 'Microwave', 'Toaster', and use a common method 'power_usage()'.
8. 8. Demonstrate hierarchical inheritance using 'Employee' base class with 'Manager' and 'Clerk' subclasses.
9. 9. Show single inheritance from 'Parent' to 'Child'. Add an attribute to child.
10. 10. Demonstrate how constructor chaining works using super().

Polymorphism

11. 1. Create two classes 'Cat' and 'Dog' with a method 'speak()'. Use a loop to call speak() on both.
12. 2. Write a function 'describe_shape(shape)' that accepts different shape objects and prints area.
13. 3. Create base class 'Employee' with method 'work()'. Override it in subclasses like 'Engineer' and 'Manager'.
14. 4. Write a function that takes a list of different objects and calls their 'display()' method.
15. 5. Use method overloading (simulate via default args) in class 'Calculator'.
16. 6. Override __str__ method in two classes for custom print behavior.
17. 7. Implement operator overloading using __add__ in a class 'Vector'.
18. 8. Create multiple subclasses of 'PaymentMethod' with unique implementations of 'pay()'.
19. 9. Write polymorphic code using duck typing (methods with same name, unrelated classes).
20. 10. Demonstrate runtime polymorphism with a parent reference calling overridden method in child class.