

Python Exercises

Exercise 1: Basic Class and Method

1. **Create a class** named `Person` with attributes `name` and `age`.
2. **Add a method** named `greet` that prints a greeting message including the person's name.
3. **Create an instance** of `Person` and call the `greet` method.

Exercise 2: Class with Initializer

1. **Create a class** named `Book` with attributes `title`, `author`, and `year`.
2. **Implement an `__init__` method** to initialize these attributes.
3. **Create an instance** of `Book` and print its attributes.

Exercise 3: Method with Return Value

1. **Create a class** named `Circle` with an attribute `radius`.
2. **Add a method** named `area` that returns the area of the circle (use the formula $\pi * \text{radius}^2$).
3. **Create an instance** of `Circle`, calculate, and print its area.

Exercise 4: Method Overloading (Using Default Arguments)

1. **Create a class** named `Calculator` with a method `add` that can take either two or three numbers.
2. **Implement** the method so that it sums up the numbers and returns the result.
3. **Create an instance** of `Calculator` and test the `add` method with both two and three numbers.

Exercise 5: Inheritance and Method Overriding

1. **Create a base class** `Vehicle` with a method `start_engine` that prints "Engine started".
2. **Create a derived class** `ElectricCar` that overrides `start_engine` to print "Electric engine started".
3. **Create instances** of both `Vehicle` and `ElectricCar` and call their `start_engine` methods.

Exercise 6: Class Method

1. **Create a class** named `Employee` with a class method `create_employee` that takes a name and salary and returns an instance of `Employee`.
2. **Add an instance method** `display` that prints the employee's name and salary.
3. **Create an instance** using the class method and call the `display` method.

Exercise 7: Static Method

1. **Create a class** named `MathUtils` with a static method `multiply` that takes two numbers and returns their product.
2. **Call the static method** without creating an instance of the class.

Exercise 8: Property Decorators

1. **Create a class** named `Rectangle` with private attributes `_width` and `_height`.

2. **Use property decorators** to define getter and setter methods for these attributes.
3. **Create an instance** of `Rectangle`, set and get its width and height using the property methods.

Exercise 9: Multiple Inheritance

1. **Create two base classes** `Person` and `Employee`, each with a method `info` that prints information about the person and the employee, respectively.
2. **Create a derived class** `Manager` that inherits from both `Person` and `Employee` and overrides the `info` method to print a combined message.
3. **Create an instance** of `Manager` and call the `info` method.

Exercise 10: Method Chaining

1. **Create a class** named `Chainable` with methods `set_value`, `add`, and `multiply`.
2. **Ensure that each method** returns `self` so that methods can be chained.
3. **Create an instance** of `Chainable`, chain multiple method calls, and print the final result.