**Basic Class and Object Exercises**

1. Create a class called Person with attributes name and age. Create an object of the class and print its attributes.
2. Add a method called greet to the Person class that prints a greeting message including the person's name.
3. Create a class called Car with attributes make, model, and year. Include a method to print out the car's details.
4. Create a class Circle with a method to compute the area. Initialize the class with the radius.
5. Create a class Rectangle with methods to compute the area and perimeter. Initialize the class with the length and width.

**Inheritance and Polymorphism Exercises**

6. Create a base class Animal with a method speak. Create two derived classes Dog and Cat that override the speak method.
7. Create a base class Shape with a method area. Create derived classes Square and Triangle that implement the area method.
8. Create a class Employee with attributes name and salary. Create a derived class Manager with an additional attribute department.
9. Create a base class Vehicle with a method drive. Create derived classes Bike and Truck that override the drive method.
10. Create a base class Bird with a method fly. Create derived classes Eagle and Penguin. Override the fly method in Penguin to indicate that penguins cannot fly.

**Encapsulation and Abstraction Exercises**

11. Create a class Account with private attributes balance. Provide public methods to deposit and withdraw money.
12. Create a class Book with private attributes title, author, and pages. Provide public methods to get and set these attributes.
13. Create a class Laptop with private attributes brand, model, and price. Provide a method to apply a discount and a method to display laptop details.
14. Create a class BankAccount with private attributes account_number and balance. Provide methods to deposit, withdraw, and check the balance.
15. Create a class Student with private attributes name, grade, and age. Provide methods to get and set these attributes and a method to display the student's details.

**Class Relationships and Advanced Concepts Exercises**

16. Create a class Library with attributes name and books (a list of Book objects). Provide methods to add and remove books.
17. Create a class School with attributes name and students (a list of Student objects). Provide methods to add and remove students.
18. Create a class Team with attributes name and members (a list of Person objects). Provide methods to add and remove members.
19. Create a class Company with attributes name and employees (a list of Employee objects). Provide methods to add and remove employees.
20. Create a class Zoo with attributes name and animals (a list of Animal objects). Provide methods to add and remove animals.

**File Handling and Exceptions Exercises**

21. Create a class FileManager with methods to read from and write to a file.
22. Create a class Log with methods to write error messages to a log file.
23. Create a class Config that reads configuration settings from a file and provides methods to access these settings.
24. Create a class Database that connects to a database and provides methods to execute queries. Handle exceptions if the connection fails.
25. Create a class Report that generates a report from data in a file. Provide methods to handle exceptions if the file does not exist or cannot be read.

**Real-world Application Exercises**

26. **Create a class Ticket for a movie theater with attributes movie_name, seat_number, and price. Provide methods to display ticket details and apply discounts.**
27. **Create a class ShoppingCart with methods to add, remove, and display items. Each item should be an object of a class Item with attributes name and price.**
28. **Create a class Restaurant with attributes name and menu (a list of Item objects). Provide methods to add and remove items from the menu.**
29. **Create a class Flight with attributes flight_number, destination, and passengers (a list of Person objects). Provide methods to add and remove passengers.**
30. **Create a class Hotel with attributes name and rooms (a list of Room objects). Each Room should have attributes room_number and is_occupied. Provide methods to book and check-out rooms.**

**GUI Application Exercises**

36. Create a class CounterApp that uses tkinter to create a simple counter GUI with increment and decrement buttons.

37. Create a class ToDoApp that uses tkinter to create a to-do list GUI where users can add and remove tasks.
38. Create a class CalculatorApp that uses tkinter to create a simple calculator GUI.
39. Create a class LoginApp that uses tkinter to create a login form GUI.
40. Create a class WeatherApp that uses tkinter to create a weather information GUI.