PYTHON

Assignment

1. **Create a class Person with attributes name and age. Write an instance method introduce that prints a message introducing the person.**
2. **Create a class Car with a class attribute total_cars. Write a class method count_cars that returns the total number of car instances created.**
3. **Create a class MathOperations with a static method add(a, b) that returns the sum of two numbers. Demonstrate its use without creating an instance.**
4. **Create two classes Employee and Department. The Employee class has attributes like name and id, and the Department class receives an Employee instance and prints the employee's details.**
5. **Create a class Library with a nested class Book. The Library class has attributes like name and location, while Book has attributes like title and author. Instantiate Book inside Library.**
6. **Create a class FileHandler that opens a file in the constructor. Implement a destructor to close the file when the object is deleted.**
7. **Create a class Resource and print a message in its destructor. Create multiple instances and observe when they are collected by Python's garbage collector.**
8. **Create a class Shape with an instance method area() and a class method default_shape(). Show how these methods behave differently.**
9. **Create a class Converter with a static method celsius_to_fahrenheit(c) to convert temperature. Show that it works without an instance.**
10. **Create two classes Student and Course. The Course class contains a method add_student to store a Student instance in a list. Test passing multiple students to the course.**
11. **Create a class Animal and subclasses Dog and Cat. Add methods in the parent and subclasses, and test how Python resolves method calls using MRO.**

12. **Create a base class Vehicle with an attribute wheels and a derived class Bicycle that inherits wheels. Modify wheels in the child class and check how it affects the base class.**
13. **Create a class Company with a class attribute company_name. Create multiple instances and show how modifying the class attribute affects all instances.**
14. **In a BankAccount class, create a nested class Address that stores the account holder's address. Demonstrate how the nested class is used.**
15. **Write a program to demonstrate Python's reference counting mechanism. Create a class and instantiate objects, then delete some references and observe when they are garbage collected.**