Experiment No: 04

Problem Statement:

Develop a web application in PHP using various concepts of object-oriented programming like Class, Object, Function, Overloading, Constructor, and Destructor

Objective:

To implement and understand advanced object-oriented programming concepts in PHP.

Theory:

Object-Oriented PHP:
Object-Oriented PHP (OOPHP) is a programming paradigm that leverages the principles of object-oriented programming (OOP) in the context of PHP (Hypertext Preprocessor). OOPHP allows developers to structure their code in a more modular and organized manner, promoting reusability and maintainability.

Object-Oriented Concepts:
Object-oriented programming revolves around four key concepts: encapsulation, inheritance, polymorphism, and abstraction. Encapsulation involves bundling data and methods that operate on that data into a single unit, known as a class. Inheritance allows a class to inherit properties and methods from another class, promoting code reuse. Polymorphism enables objects to take on multiple forms, and abstraction involves simplifying complex systems by modelling classes based on their essential features.

Defining PHP Classes:
In PHP, a class is a blueprint or a template for creating objects. It encapsulates data and the methods that operate on that data. To define a class in PHP, you use the class keyword followed by the class name. Inside the class, you declare properties to represent data and methods to perform actions.

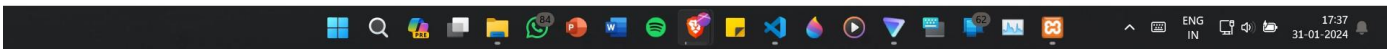
Creating Objects in PHP:
Once a class is defined, you can create objects, also known as instances, based on that class. Objects represent specific instances of the class and allow you to work with the data and methods defined within the class. To create an object in PHP, you use the new keyword followed by the class name, and you can then access properties and methods using the object instance.

Conclusion:

In this simple PHP web application, we implemented object-oriented programming concepts such as classes, objects, inheritance, function overloading, constructor, and destructor. The program defines a basic user management system with two user types: Admin and Regular User. The Admin class inherits from the base User class, showcasing method overriding. The example provides a foundation for more complex web applications where object-oriented principles can be applied for code organization.

Output (Screenshots):

A new red Toyota car created.
This is a red Toyota.
A new red Ferrari car created.
This is a red Ferrari and can speed up to 200 mph.
The red Ferrari car is being destroyed.
The red Toyota car is being destroyed.



GitHub Link:

https://github.com/NotNatrick/AWTL/tree/main/exp4