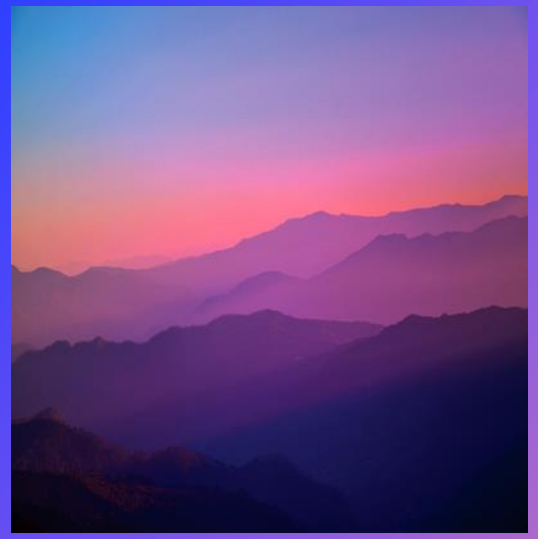
OBJECT-ORIENTED PROGRAMMING (OOP)

Christopher Ereforokuma

Introduction

Object-Oriented Programming (OOP) is a programming paradigm that organizes code around objects.



WHY OOP

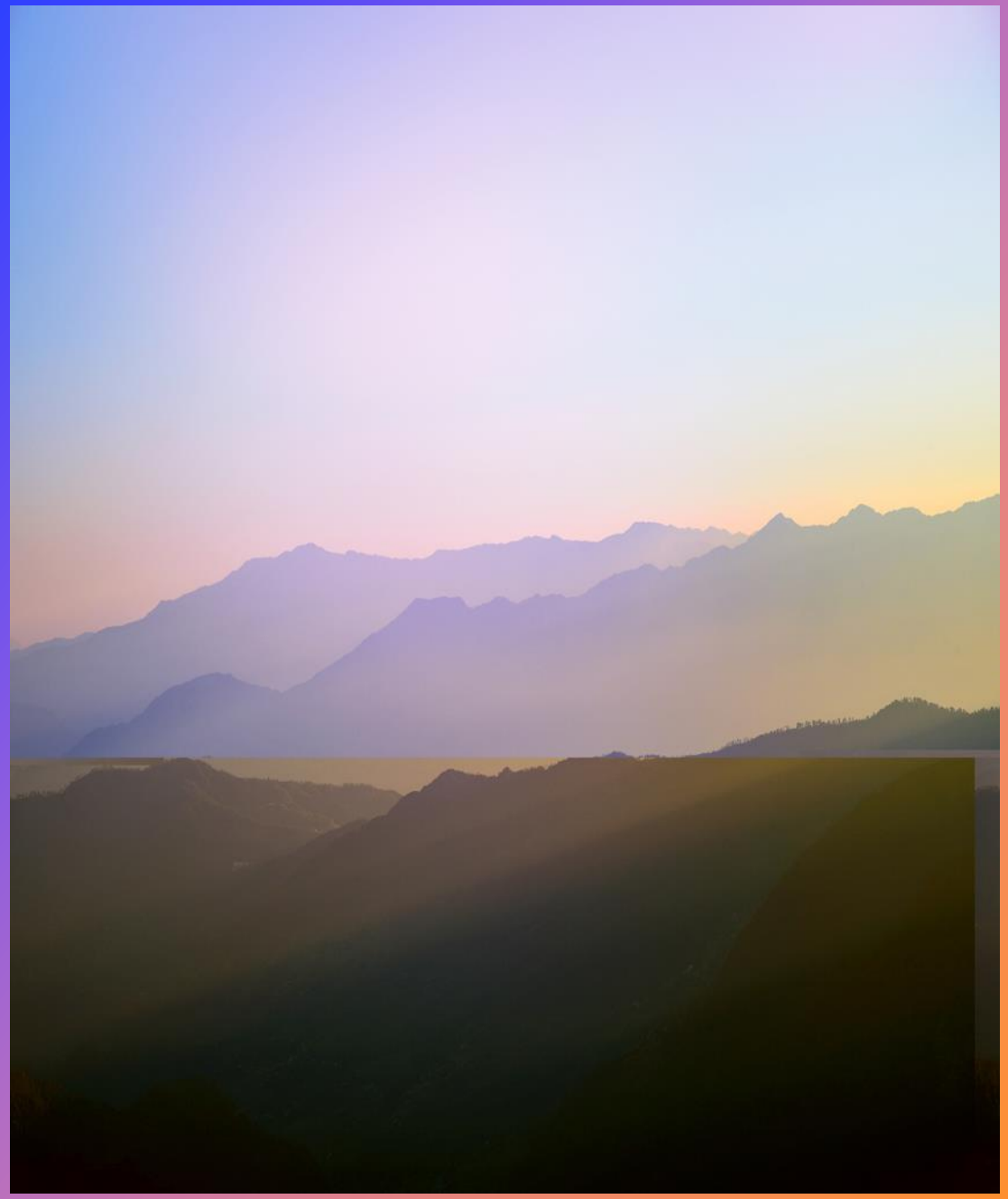
Better code reuse: OOP enables developers to create reusable code that can be used across different projects.

Improved code quality: OOP promotes better code organization, readability, and maintainability.

Easier code extensibility: OOP allows developers to add new features to existing code without breaking it.

Pillars of OOP

Encapsulation	Inheritance	Polymorphism	Abstraction
With Encapsulation, data and methods are hidden inside objects, protecting them from external interference and improving code security..	Inheritance allows classes to inherit properties and behaviors from other classes, reducing code duplication.	Polymorphism enables the use of a single interface to represent multiple types of objects, improving code flexibility.	Abstraction involves hiding complex details and showing only the essential features of an object..



CONCLUSION

OOP is a powerful programming paradigm that structures code around objects.
Its four pillars - encapsulation, inheritance, polymorphism, and abstraction - provide a structured approach to software development.
By mastering OOP, developers can create more efficient and maintainable code, build software applications that are easier to extend, debug, and scale, and ultimately, become better programmers.

THANK YOU

Christopher Ereforokuma

cereforokuma@gmail.com

https://github.com/fero-chris94