

Agenda

03 February 2025 17:57

1. Why we need to study OOP (The problem and solution)
2. Class & Object
3. Four pillars of oops
 - 3.1 Inheritance
 - 3.2 Polymorphism
 - 3.2 Encapsulation
 - 3.4 Abstraction

Why we need to study OOP (The problem and solution)

03 February 2025 18:15

The Problem:

=====

What is Procedural Programming?

Procedural programming follows a step-by-step approach, where a program is structured as a sequence of procedures (functions) that operate on data. It focuses on writing functions that perform specific tasks.

Example:

Bank application

1. ATM functions(deposit, withdraw, check balance)
2. Loan Functions(interest calc, foreclosure)

Disadvantages:

1. As features grows it is not easy to maintain
2. Need to have mappings for each feature (mapping data & functions)

The Solution:

=====

OOP:

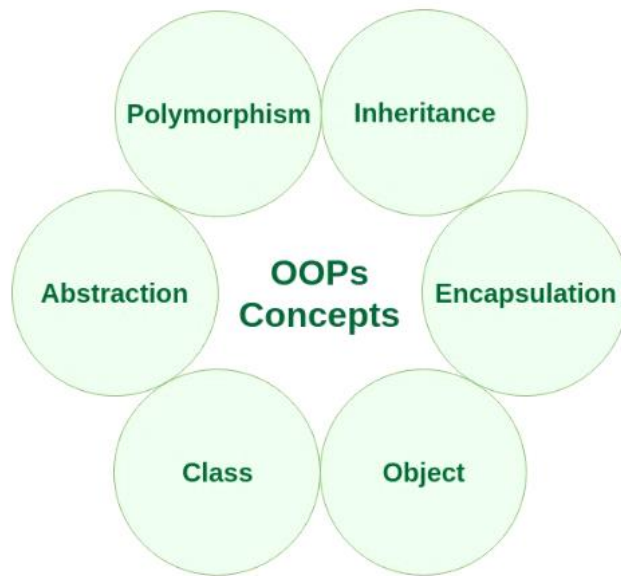
OOP organizes data(variables) and functions(behaviour) together into **objects** using **classes**.

Advantages:

Feature	Procedural Programming	Object-Oriented Programming
Structure	Uses functions and variables separately	Groups together (data and functions) into objects (classes)
Data Security	Data is global, risk of accidental modification	Data is encapsulated inside objects/class
Code Reusability	Difficult to reuse functions	Easy to reuse objects and classes
Scalability	Becomes complex handling in multiple features	Easy to extend with new features

Note: Using OOPS we can create our own data types.

OOPS pillars/principles:



Class & Object

03 February 2025 19:20

Class:

It is a blueprint and it is virtual(doesn't have memory), A class defines a set of rules/principles(attributes and methods) that object can use them.

Object:

It is an instance of class and it has physical memory, Can access those class(attributes and methods)

Examples:

Class	Objects
Car	Toyota, Suzuki, Honda
Mobile	iPhone, Samsung
Television	Sony, BPL

Let us create a banking example with below functionalities:

- Create pin
- Change pin
- Check balance
- Withdraw
- Deposit