

Difference Between Object-Oriented Programming (OOP) and Procedural Programming

Feature	Object-Oriented Programming (OOP)	Procedural Programming
Basic Unit	Object (Instance of a Class)	Procedure or Function
Data and Functions	Bundled together in Objects (Encapsulation)	Separated - Data in Structures, Functions outside
Program Structure	Organized around Objects and Classes	Organized around Procedures and Functions
Data Abstraction	Emphasizes abstraction through Classes and Objects	Less emphasis on abstraction, data may be exposed
Code Reusability	Promotes code reuse through Inheritance	Code reuse is achieved through functions and procedures
Inheritance	Hierarchical structure, allows for class hierarchy	Typically not supported, or achieved through functions
Polymorphism	Supports polymorphism, allows objects to be treated uniformly	Limited or no support for polymorphism
Encapsulation	Enforces encapsulation, hides internal details	Limited encapsulation, data may be directly accessed
Flexibility	Offers greater flexibility and adaptability	May be less flexible and modular
Example Languages	Java, C++, Python, etc.	C, Pascal, Fortran, etc.

Difference Between Object-Oriented Programming (OOP) and Procedural Programming

Related Posts:

1. Abstraction and encapsulation
2. Object Oriented Programming & Methodolog Viva Voce
3. How to install compiler for code blocks
4. Object Oriented Programming
5. Differences between Procedural and Object Oriented Programming
6. Features of Object Oriented Paradigm
7. Inheritance in Object Oriented Programming
8. Object Oriented Programming
9. Introduction to Object Oriented Thinking & Object Oriented Programming
10. features of Object oriented paradigm
11. Merits and demerits of Object Oriented methodology
12. Concept of Objects: State, Behavior & Identity of an object
13. Access modifiers
14. Static members of a Class
15. Instances in OOP
16. Message Passing in OOP
17. Construction and destruction of Objects