

OOPs Concept in C#

Agenda

- What are the Problems of Functional Programming?
- How we can overcome the Functional Programming Problem?
- What Is Object-Oriented Programming in C#?
- What are the OOPs Principles?
- Why do we need Object Oriented Programming in C#?
- Why do we need real-world objects in a Project?
- What types of programming languages come under the OOP system?

What are the Problems of Functional Programming?

Reusability:

In Functional Programming, we need to write the same code or logic at multiple places which increases the code duplication. Later if we want to change the logic, then we need to change it at all places.

Extensibility:

It is not possible in functional programming to extend the features of a function. Suppose you have a function and you want to extend it with some additional features then it is not possible. You have to create a completely new function and then change the function as per your requirement.

Simplicity:

As extensibility and reusability are not possible in functional programming, usually we end up with lots of functions and lots of scattered code.

Maintainability:

As we don't have Reusability, Extensibility, and Simplicity in functional Programming, so it is very difficult to manage and maintain the application code.

How we can overcome Functional Programming Problems?

We can overcome the functional programming problems (Reusability, Extensibility, Simplicity, and Maintainability) using Object-Oriented Programming. OOPs provide some principles and using those principles we can overcome the functional programming problems.

What Is Object-Oriented Programming?

Object-Oriented Programming (OOPs) in C# is a design approach where we think in terms of real-world objects rather than functions or methods. Unlike procedural programming language, here in oops, programs are organized around objects and data rather than action and logic.

