Roles of OOP in Large-Scale and Secure Software Design

Object-Oriented Programming (OOP) plays a crucial role in large-scale and secure software design by providing a structured and modular approach to development.

In addition to encapsulation and data hiding, inheritance and polymorphism, and abstraction, the key roles OOP plays in the below context:

1. Modularity and Reusability

2. Maintainability and Scalability

3. Security

Modularity: OOP promotes the division of a software system into smaller, manageable, and independent modules (classes). This makes the system easier to understand, develop, and maintain.
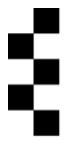
Reusability: Classes and objects can be reused across different parts of the application or even in different projects, reducing redundancy and development time.

Maintainability: OOP's modular nature makes it easier to update and maintain code. Changes in one part of the system are less likely to affect other parts.

Scalability: OOP supports the development of large-scale applications by allowing developers to build on existing code and extend functionality through inheritance and polymorphism.

Security:

- **Access Control**: OOP provides mechanisms like private and protected access modifiers to control access to sensitive data and methods, enhancing the security of the application.

- **Encapsulation**: By encapsulating data and exposing only necessary interfaces, OOP reduces the risk of unauthorised access and data corruption.