

Activity

1. What is the term for the ability of a class to have multiple methods with the same name but different parameters?
 - Overriding
 - **Overloading**
 - Inheritance
 - Encapsulation
2. What is the primary goal of encapsulation in OOP?
 - Reducing code complexity
 - Improving code reusability
 - **Separating data and behavior**
 - Enhancing code performance
3. Which OOP principle states that an object should only be modified through its methods and not directly by external code?
 - Inheritance
 - **Encapsulation**
 - Polymorphism
 - Abstraction
4. Which OOP principle promotes the idea of designing classes based on common characteristics and behaviors?
 - Encapsulation
 - **Inheritance**
 - Polymorphism
 - Abstraction
5. Which OOP principle states that a subclass should be usable in place of its superclass without affecting the correctness of the program?
 - Open/Closed Principle
 - Interface Segregation Principle
 - **Liskov Substitution Principle**
 - Dependency Inversion Principle
6. When would you prefer using an interface over an abstract class?
 - When you need to provide default implementations for some methods
 - **When you want to achieve multiple inheritance.**
 - When you need to define constants that are shared across multiple classes.
 - When you want to enforce a "is-a" relationship.

7. The Liskov Substitution Principle (LSP) is a fundamental concept for ensuring:

- Proper encapsulation of class attributes.
- The correctness of the program when using inheritance.
- All classes have a single responsibility.
- The proper use of the "super" keyword in subclasses.

8. The Interface Segregation Principle (ISP) promotes the idea of:

- Using multiple inheritance for interfaces.
- Combining multiple interfaces into a single interface.
- Defining large interfaces with many methods.
- Keeping interfaces small and focused on specific behaviors.

9. The Dependency Inversion Principle (DIP) allows for:

- Using concrete classes directly in the codebase.
- Creating tightly coupled software components.
- Simplifying the design of a software system.
- Substituting implementations of interfaces at runtime.

10. The Liskov Substitution Principle (LSP) is closely related to which other principle?

- Single Responsibility Principle (SRP)
- Open/Closed Principle (OCP)
- Interface Segregation Principle (ISP)
- Dependency Inversion Principle (DIP)

11. The Single Responsibility Principle (SRP) is primarily concerned with:

- Ensuring that methods are not duplicated in different classes.
- Keeping classes focused on a single task or responsibility.
- Using inheritance to define class relationships.
- Encapsulating data within a class.

12. Which SOLID principle helps prevent a class from becoming too large and unwieldy?

- Single Responsibility Principle (SRP)
- Open/Closed Principle (OCP)
- Liskov Substitution Principle (LSP)
- Interface Segregation Principle (ISP)

13. Violating the Liskov Substitution Principle (LSP) can lead to

- Run-time errors and incorrect program behavior
- Excessive use of inheritance in the class hierarchy
- The inability to use abstract classes.
- Inefficient memory utilization.

14. Which SOLID principle helps prevent a change in one part of the software from affecting unrelated parts?

- Single Responsibility Principle (SRP)
- Open/Closed Principle (OCP)
- Liskov Substitution Principle (LSP)
- Interface Segregation Principle (ISP)

15. Violating the Single Responsibility Principle (SRP) can lead to:

- Inconsistent class behavior.
- Large and difficult-to-maintain classes.
- Inability to use abstract classes.
- Loss of inheritance.