SOLID Design Principles

1. The “S” in SOLID stands for:

A. Simple Responsibility

B. Single Responsibility

C. Single Reactive

D. Service Responsibility

2. Which principle states that “software entities should be open for extension but closed for modification”?

A. Liskov Substitution Principle

B. Interface Segregation Principle

C. Open/Closed Principle

D. Dependency Inversion Principle

3. Which SOLID principle helps avoid “fat interfaces”?

A. Single Responsibility Principle

B. Interface Segregation Principle

C. Open/Closed Principle

D. Dependency Inversion Principle

4. Liskov Substitution Principle is mainly about:

A. A class should have only one reason to change

B. Subclasses must be replaceable for their base class

C. Modules should depend on abstractions

D. Division of interfaces

5. Which of the following violates the Liskov Substitution Principle?

A. A subclass overriding a method with compatible behavior

B. A subclass throwing unexpected exceptions

C. A subclass extending base class functionality

D. A subclass implementing an interface

6. Dependency Inversion Principle suggests that:

A. High-level modules should depend on low-level modules

B. Classes should depend on concrete classes

C. High-level and low-level modules should depend on abstractions

D. Interfaces should be avoided

7. Which of the following best describes the Single Responsibility Principle?

A. A class should extend only one class

B. A class should have only one job or responsibility

C. A class should have fewer methods

D. A class should implement only one interface

8. “Clients should not be forced to depend on interfaces they do not use” refers to:

A. LSP

B. SRP

C. ISP

D. DIP

9. The Dependency Inversion Principle promotes:

A. Tight coupling

B. Loose coupling

C. Large interfaces

D. Multiple inheritance

10. Which statement about SOLID principles is TRUE?

A. SOLID makes code harder to maintain

B. SOLID promotes tight coupling and rigid design

C. SOLID improves testability and maintainability

D. SOLID increases dependency between components

11. Which principle is violated in the following code:

```
class Invoice {
   void calculateTotal() { }
   void printInvoice() { }
   void saveToDatabase() { }
}
```

A. SRP

B. OCP

C. LSP

D. DIP

12. Which principle is being applied when a new feature is added by extending a class instead of modifying it?

A. SRP

B. OCP

C. LSP

D. ISP

13. What principle is violated in this code?

```
interface Worker {
   void work();
   void getSalary();
   void eatFood();
}
```

A. LSP

B. DIP

C. ISP

D. OCP

14. Which principle encourages “Program to an interface, not implementation”?

A. DIP

B. SRP

C. ISP

D. LSP

15. In Liskov Substitution Principle, substituting a base class with child class should:

A. Change program behavior

B. Throw new exceptions

C. Maintain correctness

D. Restrict functionality

16. Identify the principle violated:

```
class Bird {
   void fly() {}
}
class Ostrich extends Bird {
   @Override
   void fly() {
      throw new UnsupportedOperationException("Can't fly");
   }
}
```

A. SRP

B. OCP

C. LSP

D. ISP

17. Which principle suggests using smaller role-specific interfaces?

A. OCP

B. DIP

C. ISP

D. SRP

18. Code violating which principle

```
class BackendDeveloper {
   void writeJava() {}
}
class Project {
   private BackendDeveloper dev;
   Project() {
      dev = new BackendDeveloper();    }
}
```

A. DIP

B. ISP

C. SRP

D. LSP

19. What principle is violated here?

```
class Shape {
   void drawCircle() {}
   void drawSquare() {}
}
```

A. SRP

B. OCP

C. LSP

D. ISP

20. Identify the principle implemented:

```
abstract class Payment {
   abstract void pay();
}
class CardPayment extends Payment {
   void pay() {
      System.out.println("Card payment");
   }
}
```

A. SRP

B. LSP

C. DIP

D. ISP

21. Which principle encourages dependency injection?

A. DIP

B. OCP

C. SRP

D. ISP

22. What is violated here?

```
class Report {
   void generate() {}
   void exportPDF() {}
```

```
    void exportExcel() {}
}
```

A. OCP

B. SRP

C. LSP

D. DIP

23. Which principle states subclasses should extend behavior, not reduce it?

A. SRP

B. LSP

C. DIP

D. ISP

24. In which principle do we avoid modifying old code to add new logic?

A. SRP

B. OCP

C. DIP

D. ISP