

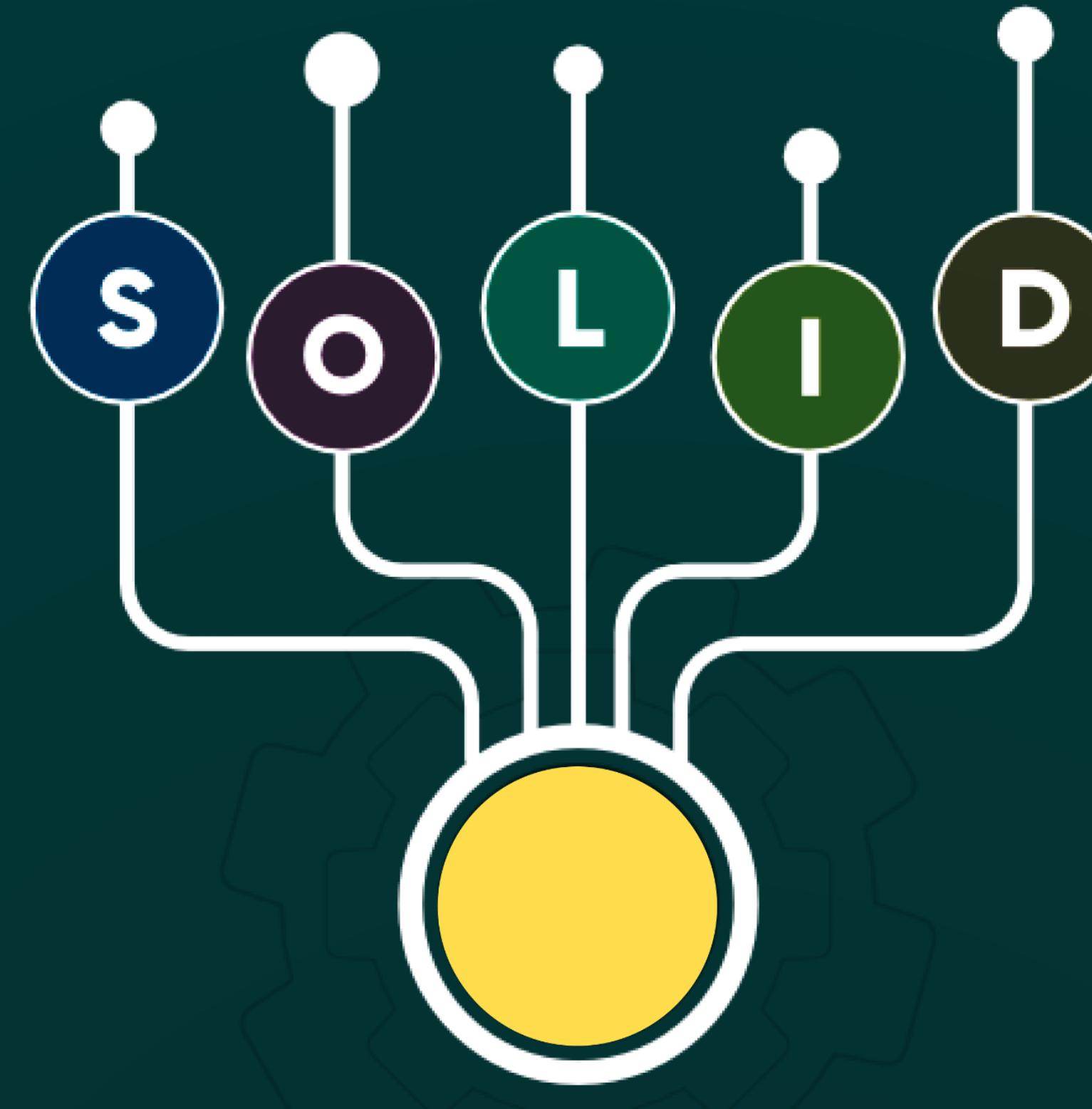
BECOME MASTER

in

SOLID PRINCIPLES

with these

Most Asked Questions

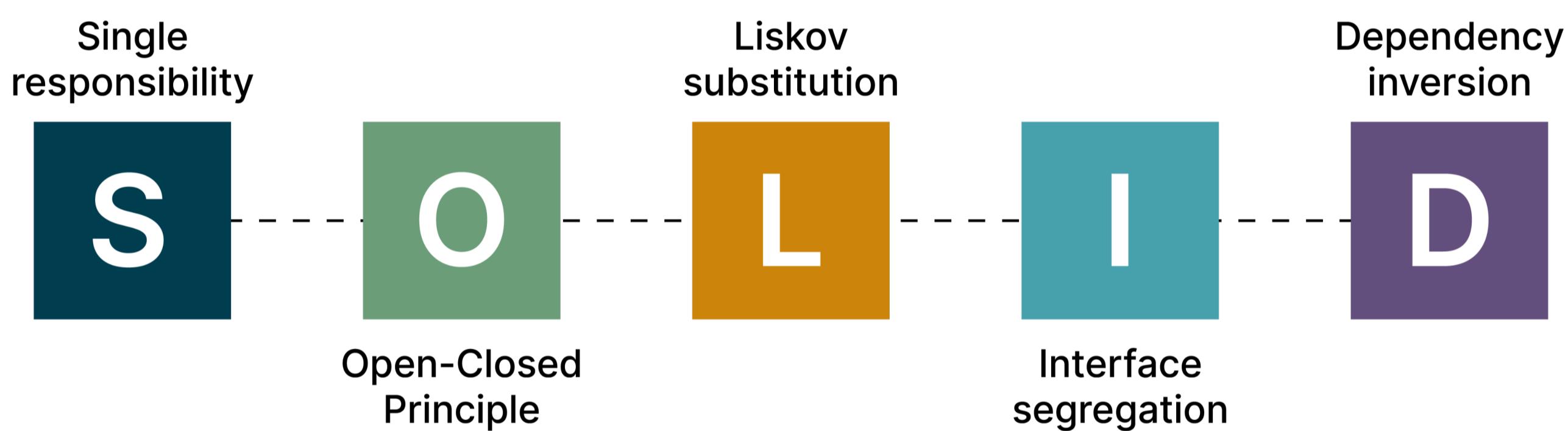


Question 1

What is the importance of Solid Principles?

The SOLID principles were developed by Robert C. Martin.

SOLID is a popular set of design principles that are used in object-oriented software development. If it is applied properly it makes your code more extendable, logical, and easier to read.



Goal -

The goal of the SOLID principles is to reduce dependencies so that engineers change one area of software without impacting others.

Read more about it -

[Importance of Solid Design Principles](#)



[Solid Principles by Nahidul](#)



Question 2

What are SOLID principles?

SOLID is an acronym that stands for five key design principles:

S

Single Responsibility Principle,

O

Open-Closed Principle

L

Liskov Substitution Principle

I

Interface Segregation Principle

D

Dependency Inversion Principle



Subhadip
Chowdhury



From



To



Placed with

100% Hike

Question 3

Describe the Single Responsibility Principle (SRP).

As the name suggests, this principle states that each class should have one responsibility, one single purpose. This means that a class will do only one job, which leads us to conclude it should have only one reason to change.

If A Class Has Many Responsibilities, It Increases The Possibility Of Bugs Because Making Changes To One Of Its Responsibilities Could Affect The Other Ones Without You Knowing.

Goal -

This principle aims to separate behaviors so that if bugs arise as a result of your change, it won't affect other unrelated behaviors.

Read more about it -

[SRP In java](#)



[SRP in Detail](#)



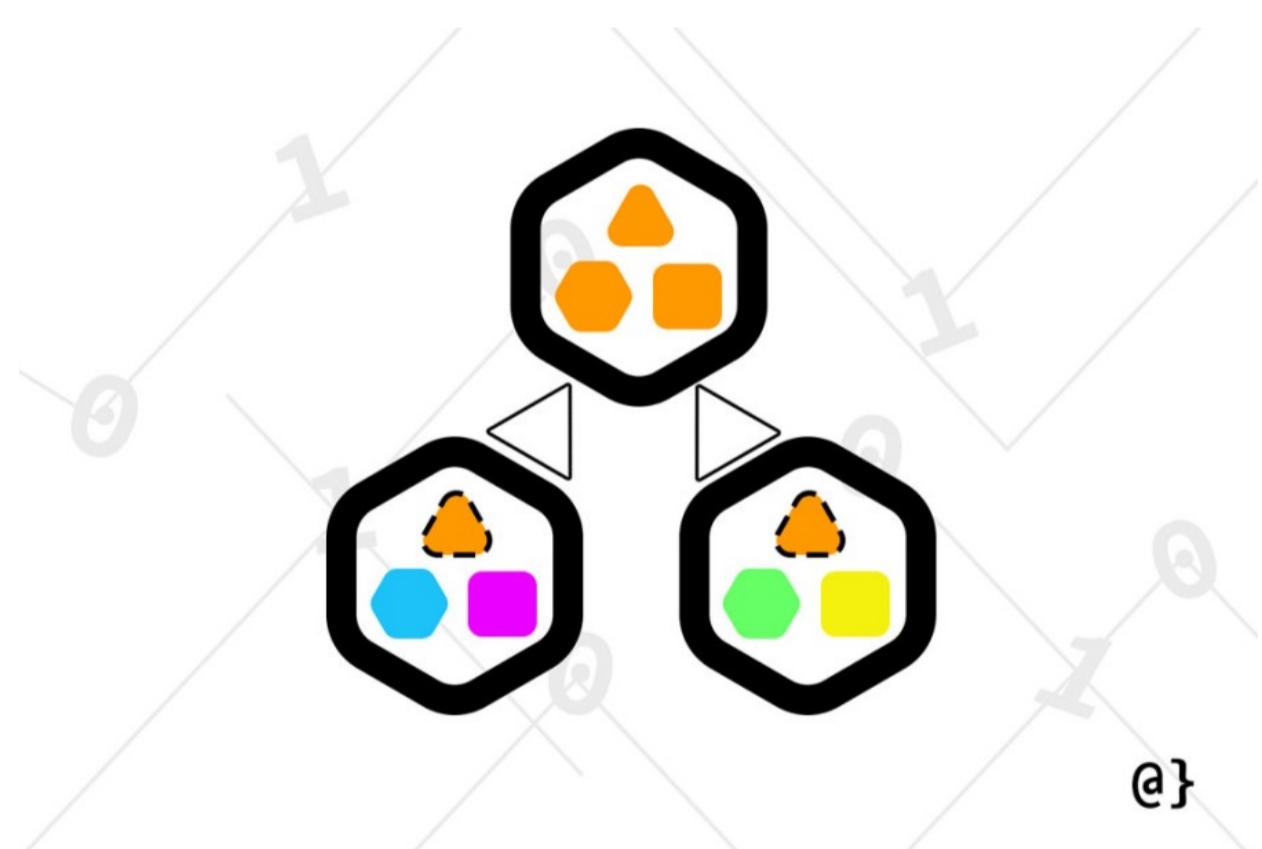
Question 4

What is open-closed principle?

According to Robert Martin, it says that:

A software artifact – such as a class or a component – should be open for extension but closed for modification.

The Open Close Principle States That The Design And Writing Of The Code Should Be Done In A Way That New Functionality Should Be Added With Minimum Changes In The Existing Code. The Design Should Be Done In A Way To Allow The Adding Of New Functionality As New Classes, Keeping As Much As Possible Existing Code Unchanged.



Resources -

[Open close principle in Java](#)



[Open close principle by aaina jain](#)



Question 5

What is the Liskov substitution principle?

Robert C. Martin summarizes it:

“Subtypes must be substitutable for their base types.”

The Liskov Substitution Principle (LSP) Is A Fundamental Principle In Object-Oriented Programming That States That Objects Of A Superclass Should Be Able To Be Replaced With Objects Of A Subclass Without Affecting The Correctness Of The Program.



Resources -

[LSP explained in detail](#)



[LSP in Java](#)



Question 6

What is the interface segregation principle?

According to Mr Robert

“Clients should not be forced to depend upon interfaces that they do not use.”

The Interface Segregation Principle (ISP) States **That A Client Should Not Be Exposed To Methods It Doesn't Need**. Declaring Methods In An Interface That The Client Doesn't Need Pollutes The Interface And Leads To A “Bulky” Or “Fat” Interface.

Resource to read more in details with example -

[ISP explained in Java](#)



[ISP explained with detail example](#)



Gopal Yadav

From



To



Placed with

210% Hike

Question 7

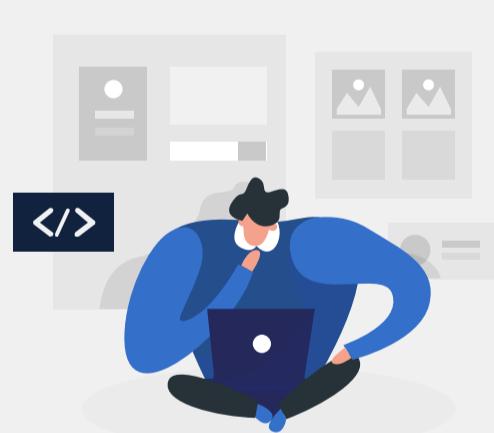
What is the dependency inversion principle?

The Dependency Inversion Principle (DIP) States That High Level Modules Should Not Depend On Low Level Modules; Both Should Depend On Abstractions.

In Object-Oriented Design, The Dependency Inversion Principle Is A Specific Methodology For Loosely Coupled Software Modules.

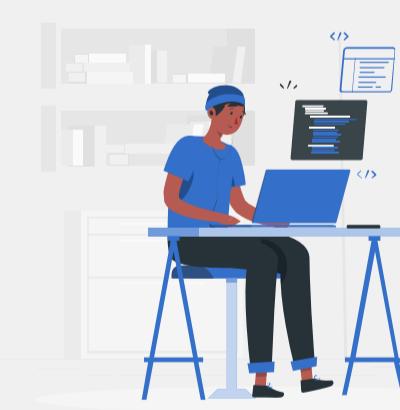
Courses Offered by Tutor

DSA with System Design



[Learn more →](#)

Full Stack with MERN



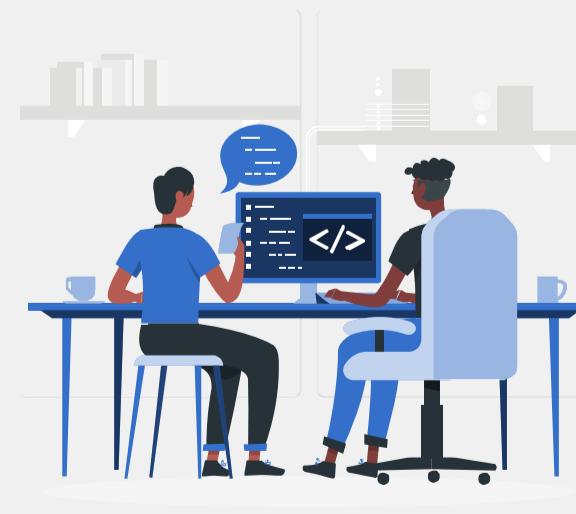
[Learn more →](#)

Data Science & Machine Learning



[Learn more →](#)

Full Stack Data Science (AI & ML)



[Learn more →](#)

Question 8

How are Design Principles different from Design Patterns?

Design Principles

Design Principles Are General (High-Level) Guidelines That Help Inform Good Software Design. You Should Have A Very Good Reason Every Time You Decide Not To Follow Principles.

Examples Of Design Principles:

- Dependency On Abstraction Not Concrete Classes
- Encapsulate That Varies
- Program To Interfaces Not Implementations

Resource to read more in details with example -

[Design Pattern vs Design principles](#)



[Importance of Solid Design Principles](#)



Design Patterns

Design Patterns Are Specific (Low-Level) Solutions To Specific Problems. You Should Have A Very Good Reason Every Time You Decide To Implement A Pattern. Design Patterns Are Already Invented Solutions Which Are Well Tested And Safe To Use.
Examples Of Design Patterns:

Examples Of Design Patterns:

- Single Design Pattern: When You Want Only One Instance Of A Class.
- Repository Design Pattern: To Separate Different Layers Of Application (Business Repository, Data Repository)



Why Tutort Academy?

100% Guaranteed Job Referrals

250+ Hiring Partners

2.1CR Highest CTC



Nikesh Bisen



I got rejected in the Amazon interview. After that, I joined Tutort Academy for DSA concepts as a working professional. They fulfilled my all requirements and that is why I am in Microsoft right now. I highly recommend Tutort Academy for professionals.



Akansha Likhdhari



When I started looking for a software development course, I found Tutort Academy completely matching my requirements. Apart from the content and live classes that they provide, the mentorship program is the cherry on the cake.

Start Your Upskilling with us

Join our Job-Oriented Programs

Explore More

www.tutort.net



[Watch us on Youtube](#)



[Read more on Quora](#)

Explore our courses



[Advanced DSA and System Design Masters Program](#)



[Full Stack Specialisation In Software Development](#)

Follow us on



Phone
+91-8712338901



E-mail
contact@tutort.net