25 IMPORTANT
OOPS CONCEPT
### INTERVIEW QUESTIONS

AlgoTutor

Q 1. What is object-oriented programming (OOPs)?

Ans: Object-Oriented Programming (OOP) is a programming paradigm that organizes code into objects, which are instances of classes.

It focuses on encapsulating data and behavior together, promoting modularity and reusability.

Q 2. What is encapsulation?

Ans: Encapsulation is the concept of bundling data (attributes) and methods (functions) that operate on that data into a single unit (class). It hides the internal details of an object and provides controlled access through methods.

Data Methods ← → Data Members

Encapsulation
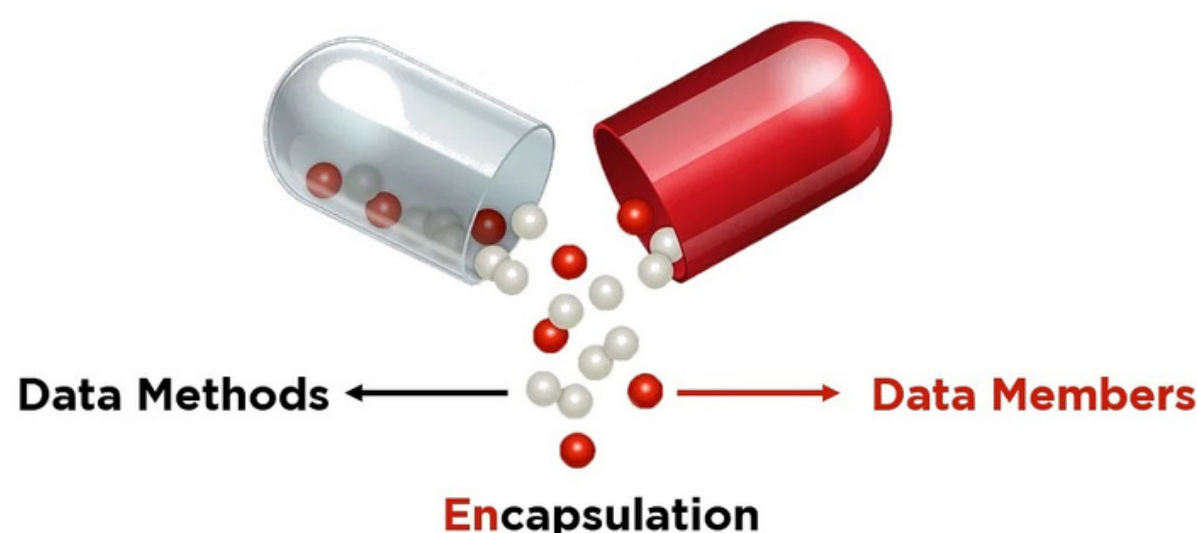
Q 3. What is inheritance?

Ans: Inheritance is a mechanism where a new class (subclass or derived class) inherits properties and behaviors from an existing class (superclass or base class).
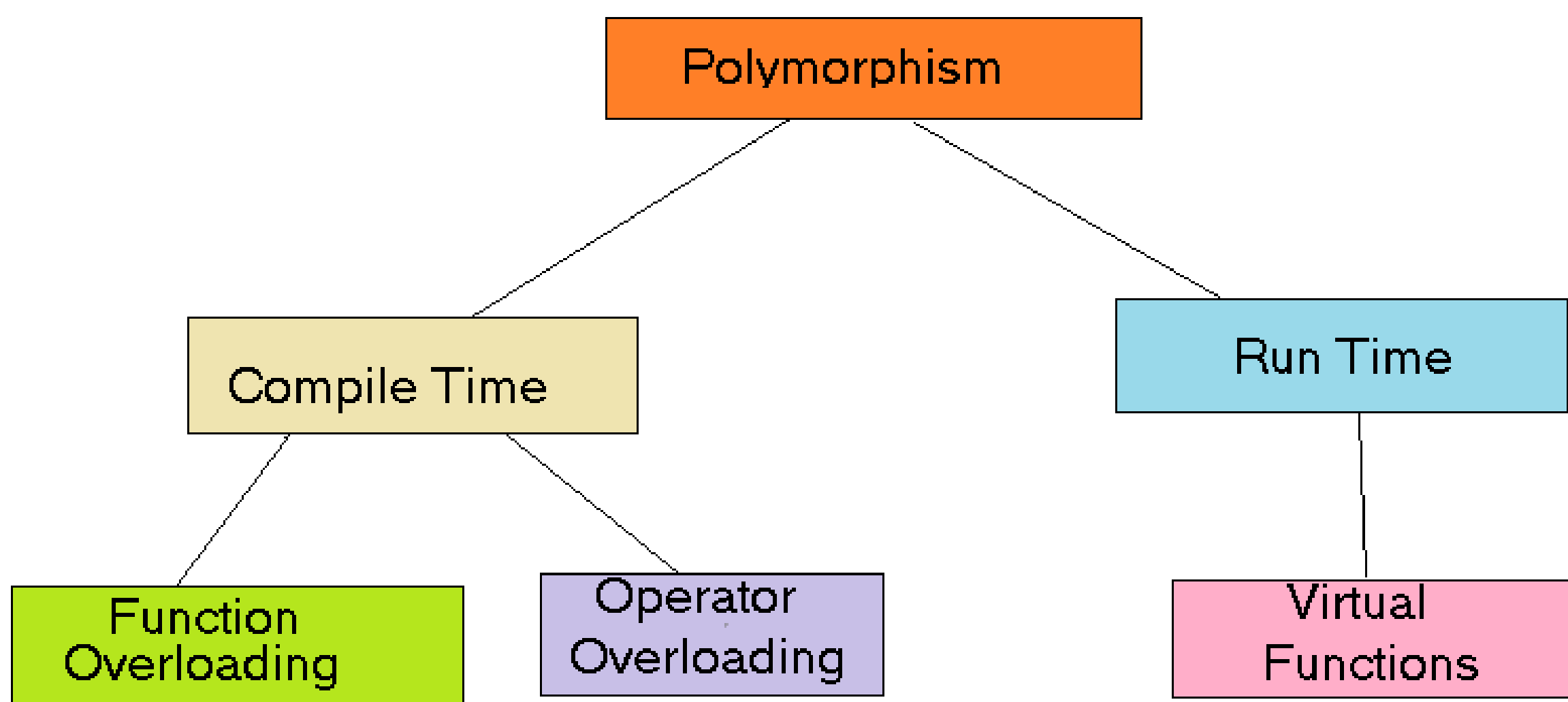
It promotes code reuse and supports hierarchical relationships.

Q 4. What is polymorphism?

Ans: Polymorphism allows objects of different classes to be treated as instances of a common superclass.

It enables methods to be invoked on objects without knowing their specific types, as long as they adhere to the common interface.

```
                        Polymorphism
                       /            \
              Compile Time          Run Time
              /          \                  \
    Function          Operator           Virtual
    Overloading       Overloading        Functions
```

Q 5. What is a constructor?

Ans: A constructor is a special method that is automatically called when an object is created.

It initializes the object's attributes and prepares the object for use. Constructors often have the same name as the class.

Q 6. What is method overloading?

Ans: Method overloading is the ability to define multiple methods in a class with the same name but different parameter lists.

The methods are differentiated based on the number or types of parameters they accept.

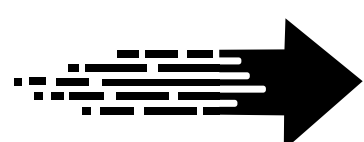## Q 7. What is method overriding?

Ans: Method overriding is the process by which a subclass provides a specific implementation for a method that is already defined in its superclass.

The overridden method in the subclass has the same name, return type, and parameters.

Q 8. What is an abstract class?

Ans: An abstract class is a class that cannot be instantiated on its own.

It may contain abstract methods (methods without a body) that must be implemented by its concrete subclasses. Abstract classes provide a common interface.

Explore Our Popular Courses

Data Structure & Algorithms

Advance System Design (LLD + HLD)

Advanced Data Science & Machine Learning

MERN Full Stack Development

Q 9. What is an interface?

Ans: An interface is a contract that defines a set of methods that a class must implement.

It allows multiple classes to adhere to the same interface, promoting a form of multiple inheritance.

Interfaces only declare method signatures, not implementations.
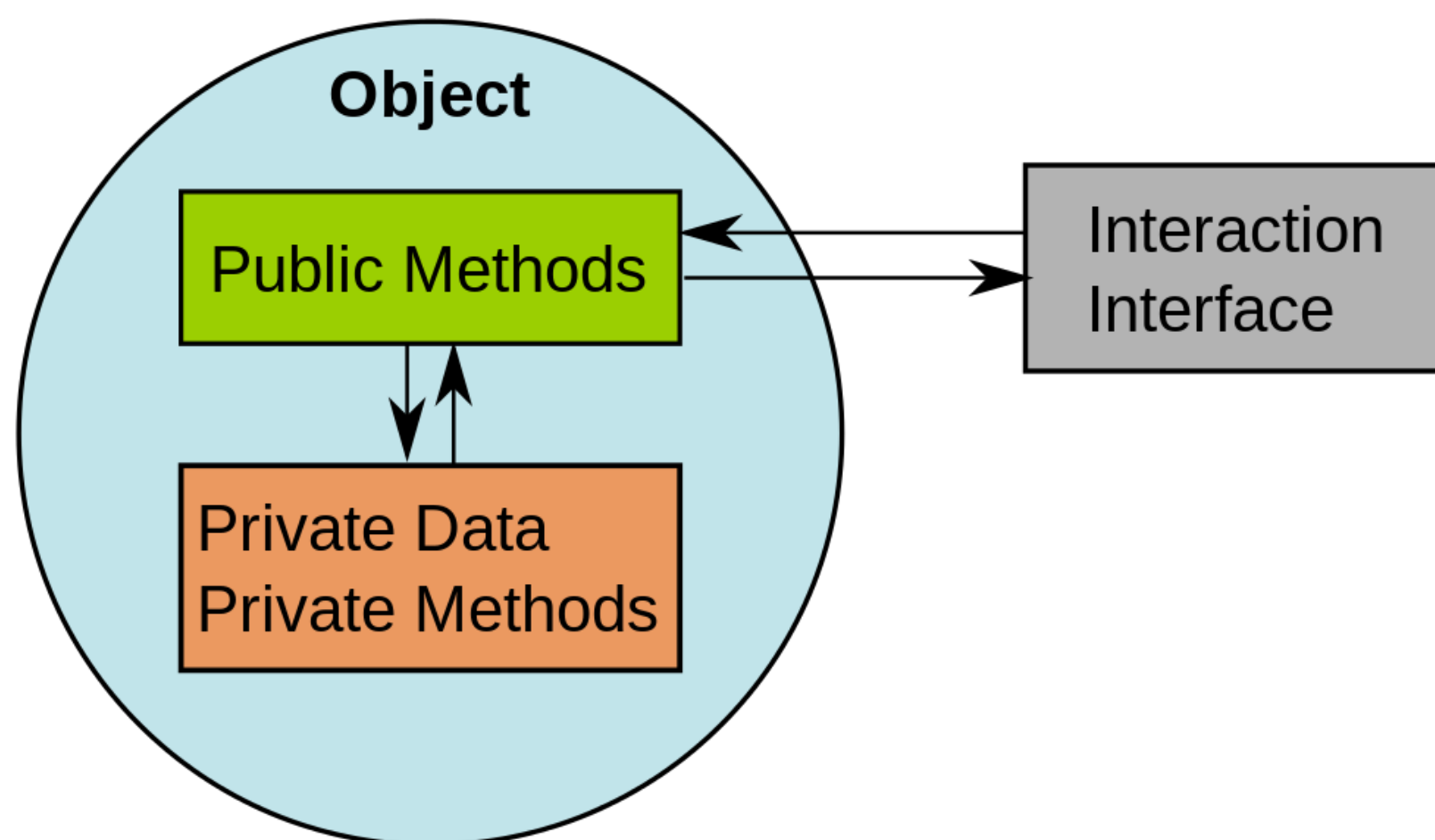
Q 10. What is a static method?

Ans: A static method belongs to the class itself, not to instances of the class.

It can be called using the class name and is often used for utility functions or operations that don't require instance-specific data.
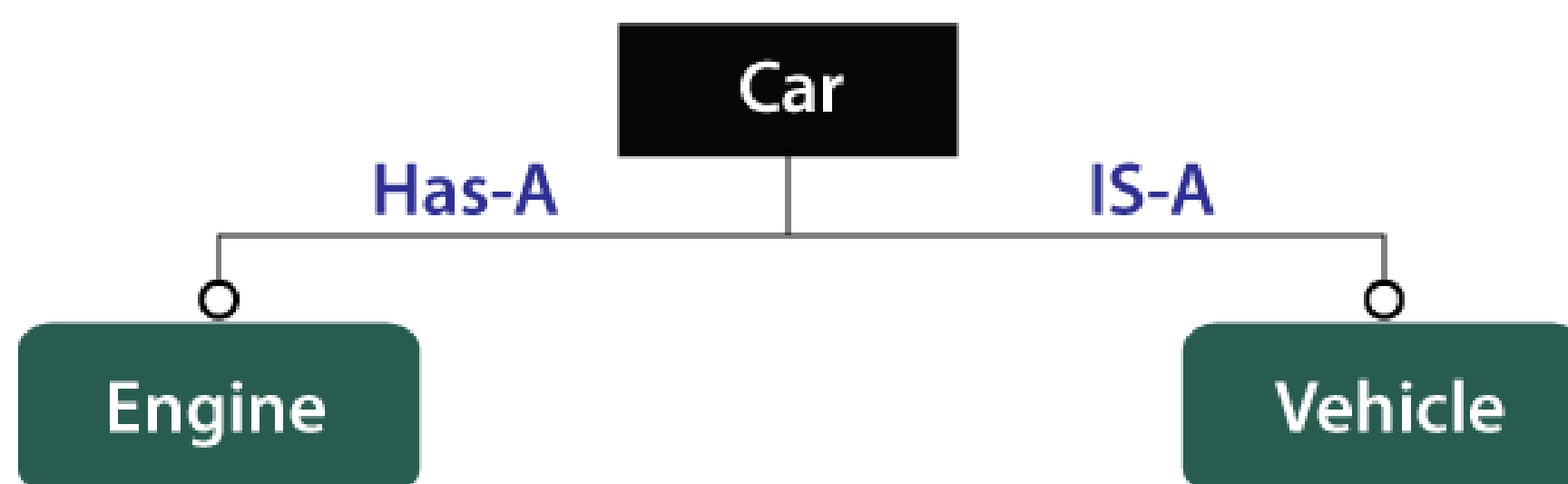
Q 11. What is a final class?

Ans: A final class is a class that cannot be subclassed. It prevents other classes from extending it and inheriting its behavior.

Q 12. What is composition?

Ans: Composition is a design principle where a class contains objects of other classes as part of its attributes.

It allows creating complex structures by combining simpler classes.

Q 13. What is a super keyword?

Ans: The **super keyword** is used to refer to the parent class or superclass. It can be used to call methods and constructors from the superclass.
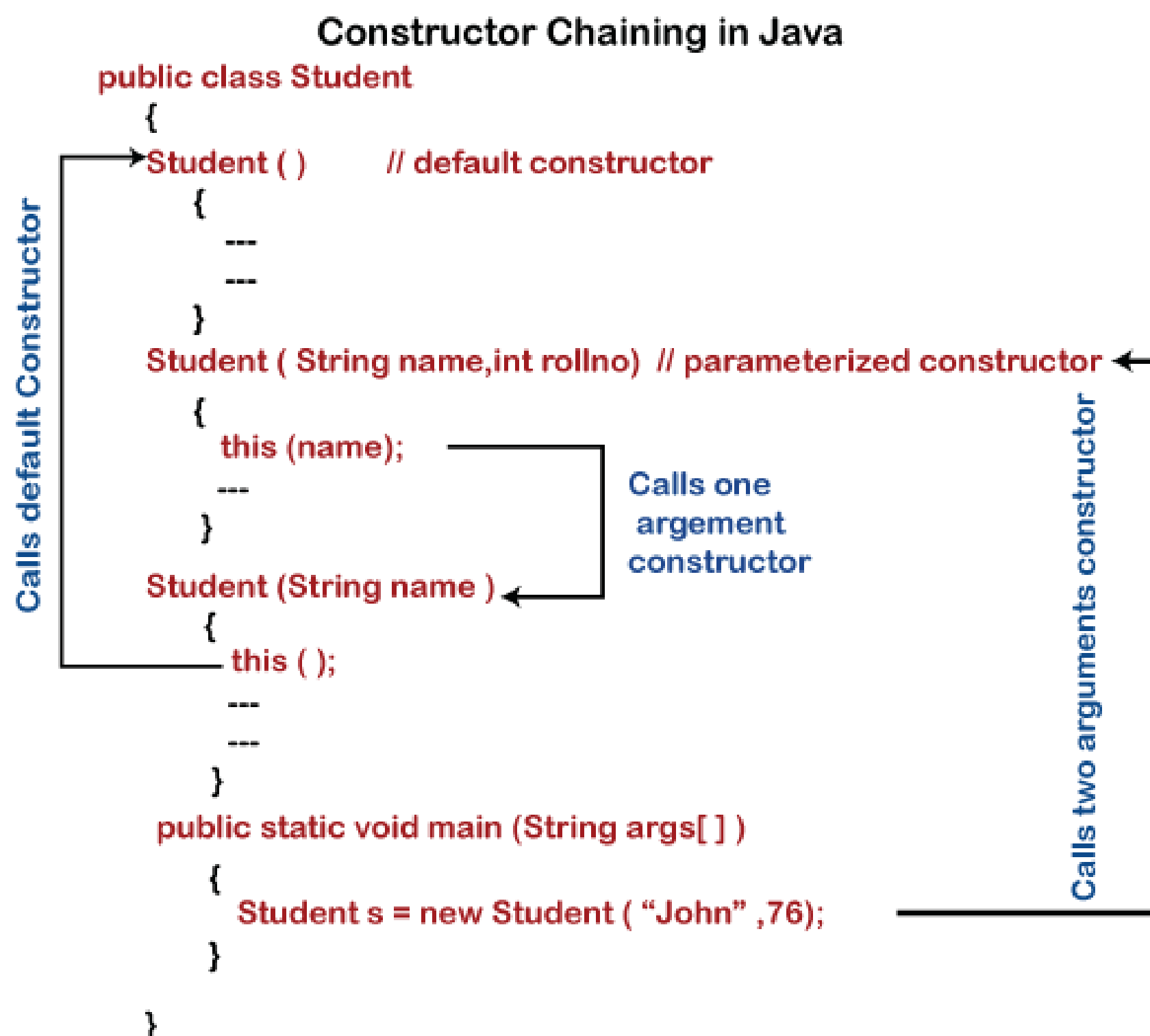
Q 14. What is method hiding?

Ans: Feature engineering involves transforming raw data into meaningful features to improve model performance.

Q 15. What is a constructor chaining?

Ans: Constructor chaining is the process of calling one constructor from another within the same class or between a superclass and a subclass.

It allows for code reuse and efficient initialization.

Constructor Chaining in Java

```
public class Student
{
    Student ( )          // default constructor
    {
        ---
        ---
    }
    Student ( String name,int rollno)  // parameterized constructor
    {
        this (name);
        ---
    }
    Student (String name )
    {
        this ( );
        ---
        ---
    }
    public static void main (String args[ ] )
    {
        Student s = new Student ( "John" ,76);
    }
}
```

Calls default Constructor

Calls one argement constructor

Calls two arguments constructor

AlgoTutor

Why Choose AlgoTutor?

100 % PLACEMENT ASSISTANCE

1:1 PERSONAL MENTORSHIP FROM INDUSTRY EXPERTS

100 % SUCCESS RATE

23 LPA(AVG.)CTC

200+ SUCCESSFUL ALUMNI

147%(AVG.) SALARY HIKE

LEARN FROM SCRATCH

CAREER SERVICES
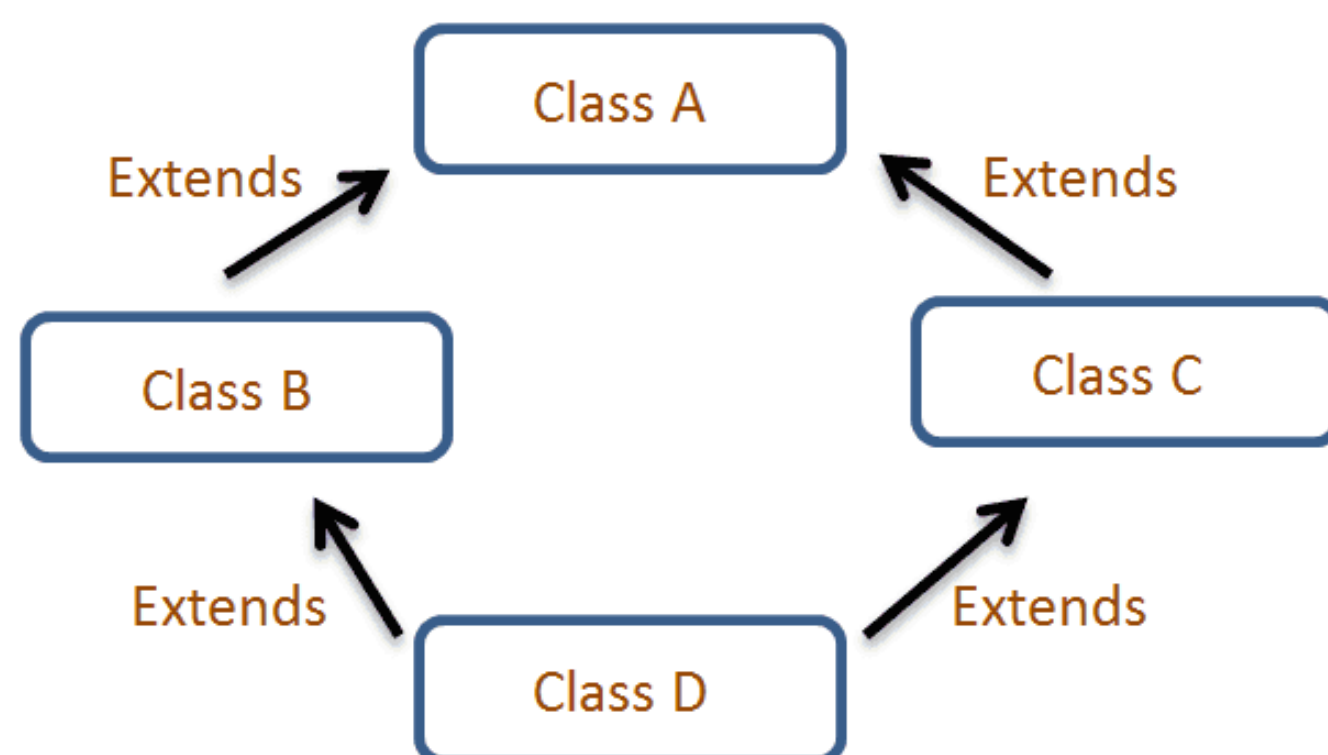
EXPLORE MORE

+91-7260058093   **www.algotutor.io**   **info@algotutor.io**

Q 16. What is the diamond problem in multiple inheritance?

Ans: The diamond problem occurs in languages that support multiple inheritance, where a class inherits from two classes that have a common base class.

This can lead to ambiguity in method resolution. Some languages provide mechanisms to handle this, like virtual inheritance.

```
                    ┌──────────┐
        Extends →   │ Class A  │   ← Extends
                    └──────────┘
   ┌──────────┐                  ┌──────────┐
   │ Class B  │                  │ Class C  │
   └──────────┘                  └──────────┘
        Extends ↖          ↗ Extends
                  ┌──────────┐
                  │ Class D  │
                  └──────────┘
```

Q 17. What is a shallow copy and a deep copy?

Ans: A shallow copy copies the references of objects contained within an object, while a deep copy creates new instances of the objects contained.

A deep copy results in a completely independent copy of the original object and its contained objects.

Q 18. What is a virtual method?

Ans: A virtual method is a method declared in a base class that can be overridden by its subclasses. The actual implementation invoked is determined by the runtime type of the object, supporting polymorphism.
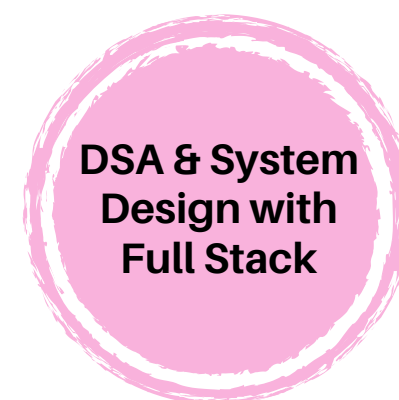
Q 19. How does Java achieve multiple inheritance?

Ans: Java achieves multiple inheritance through interfaces. A class can implement multiple interfaces, allowing it to inherit multiple sets of method declarations.

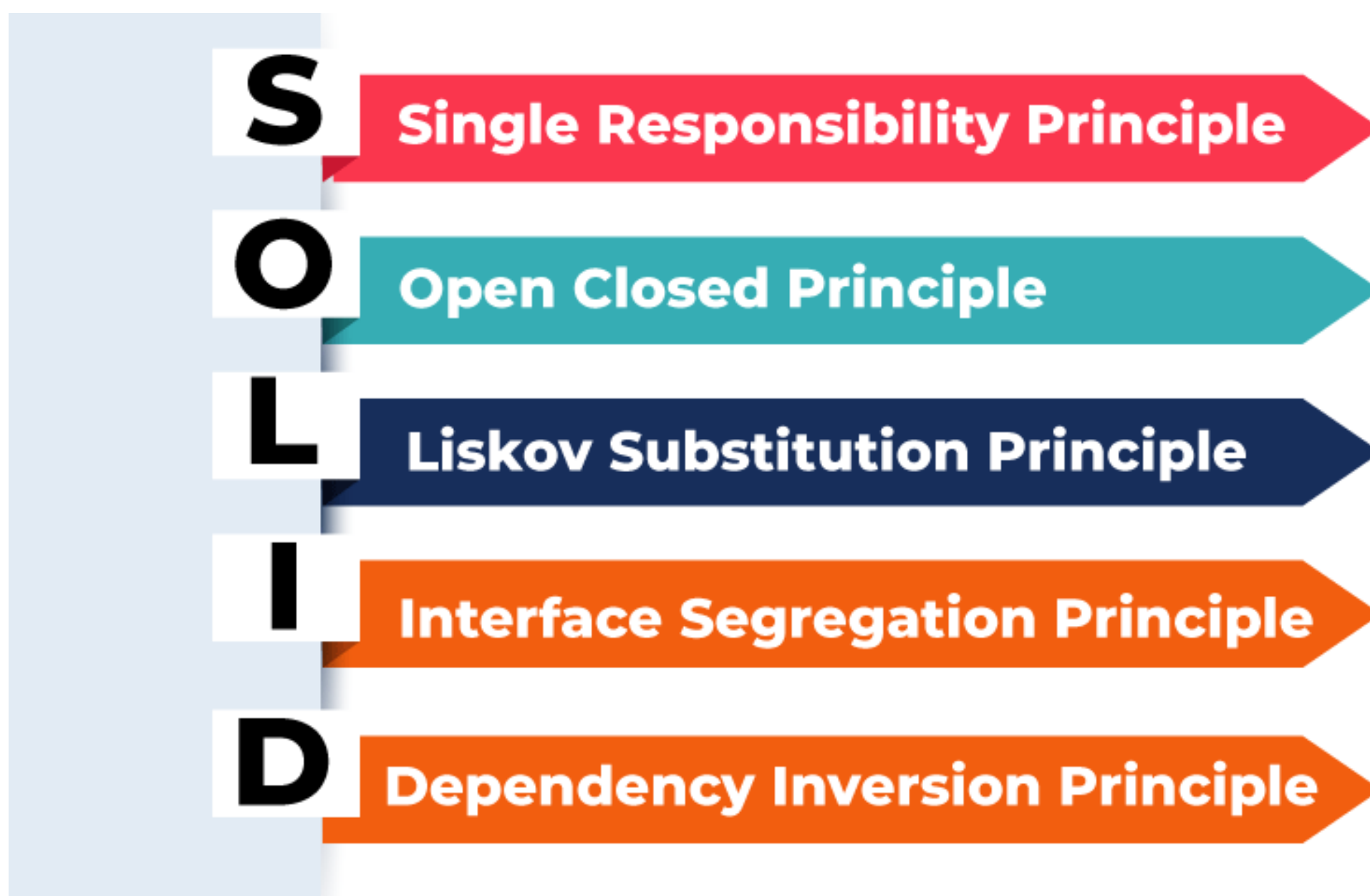## Q 20. What is the purpose of the final keyword in Java?

Ans: The final keyword can be applied to classes, methods, and variables. A final class cannot be subclassed, **a final** method cannot be overridden, and a final variable cannot be reassigned after its initial assignment.

Want to up your Skill?
Join our Popular courses

DSA with System Design (HLD + LLD)

Advanced Data Science & Machine Learning

MERN Full Stack Web Development

DSA & System Design with Full Stack

Q 21. What is the SOLID principle?

Ans: The SOLID principle is an acronym for a set of design principles that promote maintainable and scalable code:

S — Single Responsibility Principle
O — Open Closed Principle
L — Liskov Substitution Principle
I — Interface Segregation Principle
D — Dependency Inversion Principle

Q 22. What is the difference between an abstract class and an interface?

Ans: An abstract class can have both abstract and concrete methods, while an interface only has method signatures (no method implementations).

A class can implement multiple interfaces, but it can inherit from only one abstract class.

Q 23. **How is encapsulation related to data hiding?**

Ans: Encapsulation involves bundling data and methods together.

Data hiding is the practice of making the internal data of an object inaccessible to the outside world, except through well-defined methods. Encapsulation helps achieve data hiding.

Q 24. **What is the difference between composition and inheritance?**

Ans: Inheritance creates a relationship between a subclass and a superclass, allowing the subclass to inherit properties and methods.

Composition involves creating an object within another object to achieve complex behavior without the tight coupling of inheritance.

Q 25. **What is the role of a destructor in C++?**

Ans: In C++, a destructor is a special method with the same name as the class, preceded by a tilde (~). It is called when an object is about to be destroyed, allowing for cleanup tasks such as releasing resources or memory.

ABOUT US

❖ AlgoTutor is an e-learning platform that provides students with the tools and resources they need to succeed in today's competitive World.

❖ Our courses are designed by experienced educators and industry experts.

❖ We offer a variety of learning formats, including video lectures, interactive exercises, and quizzes, so you can learn at your own pace and in your own way.

❖ AlgoTutor is also committed to providing students with the support they need to succeed. Our team of tutors is available 24/7 to answer questions and provide guidance.

USP of our Programs

- 100% Placement Assistance
- 1-1 personal mentorship from Indusrty experts
- 200+ Successful Alumni
- 147(Avg.)% Salary Hike
- 100% Success Rate
- 23 LPA (Avg.) CTC
- Learn from scratch
- Career Services

Build Your Career with Us.

📈 More than **136% hike growth** for 5 out of 8 working professionals.

💵 Average package of **CTC 22 lakh.**

🧑‍🏫 Learn from **top industry** experts.

🤝 **200+** Partner Companies

Our Courses Review

Preeti Duhan

📈 90% Hike

❝ My focus and expectations was more into System Design and I have learned a lot of it, Specially High Level design and that helped me to crack Microsoft.

J.P.Morgan → ▦ Microsoft

Upendra Yadav

📈 80% Hike

❝ Consistency is the key, that I have learned from my Mentor at AlgoTutor. He usually motivates me to be consistent is class and solving problems.

COMMSCOPE' → COHESITY

Hirdesh Shriwastav

📈 160% Hike

❝ Best part of AlgoTutor was "small batch size", It was a kind of 1:1 interaction with the instructor. From all perspectives, I would definitely recommend AlgoTutor.

ORACLE → ▦ Microsoft

Shaumaya Ojha

📈 120% Hike

❝ Apart from regular live classes, I was assigned with a Mentor someone from Microsoft and he guided me very well through out the course.

J.P.Morgan → Walmart

want to
Upskill Yourself ?

Explore our Popular Courses

Data Structure & Algorithms

Advance System Design (LLD + HLD)

Advanced Data Science & Machine Learning

MERN Full Stack Development

!! Visit Our Website !!

AlgoTutor

Home   Courses ⌄   Mock Interview   Contact

Land at your dream tech company

Looking to Upskill or Career reform?

- Live interactive sessions
- Instructors are from top tech company
- 1:1 Mentorship and Doubt Solving
- Guaranteed placement support

Book a Free Demo Class

Area of Interest*

○ Software Development   ○ Data Science

Your name

Your email address

Your phone number

Need career guidance or any query? Request Callback