



Most Frequently Asked **Java**

Interview Questions

with answers



Basic Level: Page 1

Intermediate Level: Page 6

Basic Level



Join WhatsApp Group for Placement



1. What do you understand by Java virtual machine?

Answer:

JVM is a virtual machine that enables the computer to run the Java program. JVM acts like a run-time engine which calls the main method present in the Java code. JVM is the specification which must be implemented in the computer system. The Java code is compiled by JVM to be a Bytecode which is machine independent and close to the native code.

2. What is JIT compiler?

Answer:

Just-In-Time(JIT) compiler: It is used to improve the performance. JIT compiles parts of the bytecode that have similar functionality at the same time, and hence reduces the amount of time needed for compilation. Here the term “compiler” refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

3. What are the main differences between the Java platform and other platforms?

Answer:

There are the following differences between the Java platform and other platforms.

- Java is the software-based platform whereas other platforms may be the hardware platforms or software-based platforms.
- Java is executed on the top of other hardware platforms whereas other platforms can only have the hardware components.

4. What gives Java its 'write once and run anywhere' nature?

Answer:

The bytecode. Java compiler converts the Java programs into the class file (Byte Code) which is the intermediate language between source code and machine code. This bytecode is not platform specific and can be executed on any computer.

5. Is Empty .java file name a valid source file name?

Answer:

Yes, Java allows to save our java file by java only, we need to compile it by javac.java and run by java class name.

6. What are the various access specifiers in Java?

Answer:

In Java, access specifiers are the keywords which are used to define the access scope of the method, class, or a variable. In Java, there are four access specifiers given below.

- Public:- The classes, methods, or variables which are defined as public, can be accessed by any class or method.
- Protected:- Protected can be accessed by the class of the same package, or by the sub-class of this class, or within the same class.
- Default:- Default are accessible within the package only. By default, all the classes, methods, and variables are of default scope.
- Private:- The private class, methods, or variables defined as private can be accessed within the class only.

7. Difference between Heap and Stack Memory in Java. And how java utilizes this.

Answer:

Stack memory is the portion of memory that was assigned to every individual program. And it was fixed. On the other hand, Heap memory is the portion that was not allocated to the java program but it will be available for use by the java program when it is required, mostly during the runtime of the program.



Follow us on Instagram



8. How is Java different from C++?

Answer:

- C++ is only a compiled language, whereas Java is compiled as well as an interpreted language.
- Java programs are machine-independent whereas a C++ program can run only in the machine in which it is compiled.
- C++ allows users to use pointers in the program. Whereas java doesn't allow it. Java internally uses pointers.
- C++ supports the concept of Multiple inheritances whereas Java doesn't support this. And it is due to avoiding the complexity of name ambiguity that causes the diamond problem.

9. What do you understand by an instance variable and a local variable?

Answer:

Instance variables are those variables that are accessible by all the methods in the class. They are declared outside the methods and inside the class. These variables describe the properties of an object and remain bound to it at any cost.

Proud to be featured in more than **70 news articles**

What are the various access specifiers in Java?

SIGN IN SUBSCRIBE

Answer: Ed-tech platform Talent Battle crosses 3.5 Lakh registered students mark! The journey from 35 to 3.5 Lakh students! Java, access specifiers are the keywords which are used to define the scope of the method, class, or a variable. In Java, there are four access specifiers below.

February 13, 2023 19:00 IST | ANI Press Release





Join WhatsApp Group for Placement

All the objects of the class will have their copy of the variables for utilization. If any modification is done on these variables, then only that instance will be impacted by it, and all other class instances continue to remain unaffected.

Local variables are those variables present within a block, function, or constructor and can be accessed only inside them. The utilization of the variable is restricted to the block scope. Whenever a local variable is declared inside a method, the other class methods don't have any knowledge about the local variable.

10. What are the default values assigned to variables and instances in java?

Answer:

- There are no default values assigned to the variables in java. We need to initialize the value before using it. Otherwise, it will throw a compilation error of (Variable might not be initialized).
- But for instance, if we create the object, then the default value will be initialized by the default constructor depending on the data type.
- If it is a reference, then it will be assigned to null.
- If it is numeric, then it will assign to 0.
- If it is a boolean, then it will be assigned to false. Etc.

11. Why is the main method static in Java?

Answer:

The main method is always static because static members are those methods that belong to the classes, not to an individual object. So if the main method will not be static then for every object, It is available. And that is not acceptable by JVM. JVM calls the main method based on the class name itself. Not by creating the object. Because there must be only 1 main method in the java program as the execution starts from the main method. So for this reason the main method is static.



Get free mentorship
from experts



12. Can the static methods be overridden?

Answer:

- No! Declaration of static methods having the same signature can be done in the subclass but run time polymorphism cannot take place in such cases.
- Overriding or dynamic polymorphism occurs during the runtime, but the static methods are loaded and looked up at the compile time statically. Hence, these methods can't be overridden.

13. What is a ClassLoader?

Answer:

Java Classloader is the program that belongs to JRE (Java Runtime Environment). The task of ClassLoader is to load the required classes and interfaces to the JVM when required.

14. Can you tell the difference between equals() method and equality operator (==) in Java?

Answer:

We are already aware of the (==) equals operator. That we have used this to compare the equality of the values. But when we talk about the terms of object-oriented programming, we deal with the values in the form of objects. And this object may contain multiple types of data. So using the (==) operator does not work in this case. So we need to go with the .equals() method.

Both [(==) and .equals()] primary functionalities are to compare the values, but the secondary functionality is different.

15. Briefly explain the concept of constructor overloading

Answer:

Constructor overloading is the process of creating multiple constructors in the class consisting of the same name with a difference in the constructor parameters. Depending upon the number of parameters and their corresponding types, distinguishing of the different types of constructors is done by the compiler.



Follow us on Instagram

Intermediate Level



16. When can you use super keyword?

Answer:

- The super keyword is used to access hidden fields and overridden methods or attributes of the parent class.
- Following are the cases when this keyword can be used:
 1. Accessing data members of parent class when the member names of the class and its child subclasses are same. For example - We have used mathematical functions in the java program like - max(), min(), sqrt(), pow(), etc. And if we notice that, then we will find that we call it directly with the class name. Like - Math.max(), Math.min(), etc. So that is a static method. And Similarly static variables we have used like (length) for the array to get the length. So that is the static method.
 2. Static classes - A class in the java program cannot be static except if it is the inner class. If it is an inner static class, then it exactly works like other static members of the class.
 3. To call the default and parameterized constructor of the parent class inside the child class.
 4. Accessing the parent class methods when the child classes have overridden them.

17. Difference between static methods, static variables, and static classes in java.

Answer:

Static Methods and Static variables are those methods and variables that belong to the class of the java program, not to the object of the class. This gets memory where the class is loaded. And these can directly be called with the help of class names.

**Get a Free Mentorship from
experts for your Campus
Placement Preparation**

- Discuss your queries with experts
- Get a roadmap for your placement preparation



Click to know more



18. What are the advantages of Packages in Java?

Answer:

There are various advantages of defining packages in Java.

- Packages avoid the name clashes.
- The Package provides easier access control.
- We can also have the hidden classes that are not visible outside and used by the package.
- It is easier to locate the related classes.

19. What is object-oriented paradigm?

Answer:

It is a programming paradigm based on objects having data and methods defined in the class to which it belongs. Object-oriented paradigm aims to incorporate the advantages of modularity and reusability. Objects are the instances of classes which interact with one another to design applications and programs. There are the following features of the object-oriented paradigm.

Follows the bottom-up approach in program design.

- Focus on data with methods to operate upon the object's data
- Includes the concept like Encapsulation and abstraction which hides the complexities from the user and show only functionality.
- Implements the real-time approach like inheritance, abstraction, etc.

20. What is the constructor?

Answer:

The constructor can be defined as the special type of method that is used to initialize the state of an object. It is invoked when the class is instantiated, and the memory is allocated for the object. Every time, an object is created using the new keyword, the default constructor of the class is called. The name of the constructor must be similar to the class name. The constructor must not have an explicit return type..

21. How many types of constructors are used in Java?

Answer:

Based on the parameters passed in the constructors, there are two types of constructors in Java.

- **Default Constructor:** default constructor is the one which does not accept any value. The default constructor is mainly used to initialize the instance variable with the default values. It can also be used for performing some useful task on object creation. A default constructor is invoked implicitly by the compiler if there is no constructor defined in the class.
- **Parameterized Constructor:** The parameterized constructor is the one which can initialize the instance variables with the given values. In other words, we can say that the constructors which can accept the arguments are called parameterized constructors.
- **No Argument Constructor:** If a constructor does not accept any parameters, it is known as a no-argument constructor. Here we can define the constructor. When No-Argument constructor is invoked while creating the object of the class the control will pass to it.

22. What is the static variable?

Answer:

The static variable is used to refer to the common property of all objects (that is not unique for each object), e.g., The company name of employees, college name of students, etc. Static variable gets memory only once in the class area at the time of class loading. Using a static variable makes your program more memory efficient (it saves memory). Static variable belongs to the class rather than the object.

23. What is the static method?

Answer:

- A static method belongs to the class rather than the object.
- There is no need to create the object to call the static methods.
- A static method can access and change the value of the static variable.



Follow us on Instagram

24. What are the restrictions that are applied to the Java static methods?

Answer:

Two main restrictions are applied to the static methods.

- The static method cannot use non-static data member or call the non-static method directly.
- this and super cannot be used in static context as they are non-static.

25. Can we make constructors static?

Answer:

As we know that the static context (method, block, or variable) belongs to the class, not the object. Since Constructors are invoked only when the object is created, there is no sense to make the constructors static. However, if you try to do so, the compiler will show the compiler error.

Talent Battle Masterclass - Placement Reports

Highest CTC

40 LPA

85% students placed with CTC more than 6 LPA

Average CTC

8.5 LPA

Referral Opportunities
in Top Companies & Startups
by Talent Battle

Average Offers per student:

2.7

Students from
5000+ colleges
use Masterclass for
Placement Preparation



Join WhatsApp Group for Placement



26. Can we declare the static variables and methods in an abstract class?

Answer:

Yes, we can declare static variables and methods in an abstract method. As we know that there is no requirement to make the object to access the static context, therefore, we can access the static context declared inside the abstract class by using the name of the abstract class.

27. What is this keyword in java?

Answer:

The this keyword is a reference variable that refers to the current object. There are the various uses of this keyword in Java. It can be used to refer to current class properties such as instance methods, variable, constructors, etc. It can also be passed as an argument into the methods or constructors. It can also be returned from the method as the current class instance.

28. What are the main uses of this keyword?

Answer:

There are the following uses of this keyword.

- this can be used to refer to the current class instance variable.
- this can be used to invoke current class method (implicitly)
- this() can be used to invoke the current class constructor.
- this can be passed as an argument in the method call.
- this can be passed as an argument in the constructor call.
- this can be used to return the current class instance from the method.

29. How can constructor chaining be done using this keyword?

Answer:

Constructor chaining enables us to call one constructor from another constructor of the class with respect to the current class object. We can use this keyword to perform constructor chaining within the same class.

30. What are the advantages of passing this into a method instead of the current class object itself?

Answer:

As we know, that this refers to the current class object, therefore, it must be similar to the current class object. However, there can be two main advantages of passing this into a method instead of the current class object.

- this is a final variable. Therefore, this cannot be assigned to any new value whereas the current class object might not be final and can be changed.
- this can be used in the synchronized block.

31. What is the Inheritance?

Answer:

Inheritance is a mechanism by which one object acquires all the properties and behavior of another object of another class. It is used for Code Reusability and Method Overriding. The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also. Inheritance represents the IS-A relationship which is also known as a parent-child relationship.

There are five types of inheritance in Java.

- Single-level inheritance
- Multi-level inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance



Follow us on Instagram



32. Why is Inheritance used in Java?

Answer:

There are various advantages of using inheritance in Java that is given below.

- Inheritance provides code reusability. The derived class does not need to redefine the method of base class unless it needs to provide the specific implementation of the method.
- Runtime polymorphism cannot be achieved without using inheritance.
- We can simulate the inheritance of classes with the real-time objects which makes OOPs more realistic.
- Inheritance provides data hiding. The base class can hide some data from the derived class by making it private.
- Method overriding cannot be achieved without inheritance. By method overriding, we can give a specific implementation of some basic method contained by the base class.

Complete Masterclass Courses and Features

- | | | |
|---|---|--|
| <ul style="list-style-type: none">• Aptitude Cracker Course• C Programming• C++• Java• Python• Data Structures and Algorithms• Operating Systems• Computer Networks• DBMS• Topic-wise Mock Tests | <ul style="list-style-type: none">• Company Wise Mock Tests• Technical & Personal Interview Preparation• One to One Mock Interviews• Full Stack Development• Artificial Intelligence• Machine Learning• Data Analytics• Data Science• PowerBI• Tableau | <ul style="list-style-type: none">• Real-Time projects on AI, ML, & Data Science• Mini Projects based on C, C++, Java, Python• TCS iON Remote Internship (For 2 years course)• Certifications by Talent Battle and TCS iON• LIVE Lectures + Recorded Courses |
|---|---|--|

Complete Masterclass Courses and Features

TCS NQT | Accenture | Capgemini | Cognizant | Infosys | Wipro | Tech Mahindra | LTI | DXC

Hexaware | Persistent | Deloitte | Mindtree | Virtusa | Goldman Sachs | Bosch | Samsung Amazon | Nalsoft | Zoho Cisco and 10+ more companies preparation.



33. Why is multiple inheritance not supported in java?

Answer:

To reduce the complexity and simplify the language, multiple inheritance is not supported in java. Consider a scenario where A, B, and C are three classes. The C class inherits A and B classes. If A and B classes have the same method and you call it from child class object, there will be ambiguity to call the method of A or B class.

Since the compile-time errors are better than runtime errors, Java renders compile-time error if you inherit 2 classes. So whether you have the same method or different, there will be a compile time error.

34. What is super in java?

Answer:

The super keyword in Java is a reference variable that is used to refer to the immediate parent class object. Whenever you create the instance of the subclass, an instance of the parent class is created implicitly which is referred by super reference variable. The super() is called in the class constructor implicitly by the compiler if there is no super or this.

35. What are the main uses of the super keyword?

Answer:

There are the following uses of super keyword.

- super can be used to refer to the immediate parent class instance variable.
- super can be used to invoke the immediate parent class method.
- super() can be used to invoke immediate parent class constructor.

36. What is method overloading?

Answer:

Method overloading is the polymorphism technique which allows us to create multiple methods with the same name but different signature. We can achieve method overloading in two ways.

- By Changing the number of arguments
- By Changing the data type of arguments

Method overloading increases the readability of the program. Method overloading is performed to figure out the program quickly.

37. What is method overriding?

Answer:

If a subclass provides a specific implementation of a method that is already provided by its parent class, it is known as Method Overriding. It is used for runtime polymorphism and to implement the interface methods.

Rules for Method overriding

- The method must have the same name as in the parent class.
- The method must have the same signature as in the parent class.
- Two classes must have an IS-A relationship between them.

38. Can we change the scope of the overridden method in the subclass?

Answer:

Yes, we can change the scope of the overridden method in the subclass. However, we must notice that we cannot decrease the accessibility of the method. The following point must be taken care of while changing the accessibility of the method.

- The private can be changed to protected, public, or default.
- The protected can be changed to public or default.
- The default can be changed to public.
- The public will always remain public.



Follow us on Instagram



39. Can we modify the throws clause of the superclass method while overriding it in the subclass?

Answer:

Yes, we can modify the throws clause of the superclass method while overriding it in the subclass. However, there are some rules which are to be followed while overriding in case of exception handling.

- If the superclass method does not declare an exception, subclass overridden method cannot declare the checked exception, but it can declare the unchecked exception.
- If the superclass method declares an exception, subclass overridden method can declare same, subclass exception or no exception but cannot declare parent exception.

40. What is the final variable?

Answer:

In Java, the final variable is used to restrict the user from updating it. If we initialize the final variable, we can't change its value. In other words, we can say that the final variable once assigned to a value, can never be changed after that. The final variable which is not assigned to any value can only be assigned through the class constructor.

Some of our placed students from Complete Masterclass



Shrinija Kalluri
Placed in Oracle
9 LPA

Placed in Accenture
6.5 LPA



Join WhatsApp Group for Placement

41. How would you differentiate between a String, StringBuffer, and a StringBuilder?

Answer:

Storage area: In string, the String pool serves as the storage area. For StringBuilder and StringBuffer, heap memory is the storage area.

Mutability: A String is immutable, whereas both the StringBuilder and StringBuffer are mutable.

Efficiency: It is quite slow to work with a String. However, StringBuilder is the fastest in performing operations. The speed of a StringBuffer is more than a String and less than a StringBuilder. (For example appending a character is fastest in StringBuilder and very slow in String because a new memory is required for the new String with appended character.)

Thread-safe: In the case of a threaded environment, StringBuilder and StringBuffer are used whereas a String is not used. However, StringBuilder is suitable for an environment with a single thread, and a StringBuffer is suitable for multiple threads.

42. Using relevant properties highlight the differences between interfaces and abstract classes.

Answer:

Availability of methods: Only abstract methods are available in interfaces, whereas non-abstract methods can be present along with abstract methods in abstract classes.

Variable types: Static and final variables can only be declared in the case of interfaces, whereas abstract classes can also have non-static and non-final variables.

Inheritance: Multiple inheritances are facilitated by interfaces, whereas abstract classes do not promote multiple inheritances.

Data member accessibility: By default, the class data members of interfaces are of the public-type. Conversely, the class members for an abstract class can be protected or private also.

Implementation: With the help of an abstract class, the implementation of an interface is easily possible. However, the converse is not true;

43. What is a Comparator in java?

Answer:

Consider the example where we have an ArrayList of employees like (EId, Ename, Salary), etc. Now if we want to sort this list of employees based on the names of employees. Then that is not possible to sort using the Collections.sort() method. We need to provide something to the sort() function depending on what values we have to perform sorting. Then in that case a comparator is used.

Comparator is the interface in java that contains the compare method. And by overloading the compare method, we can define that on what basis we need to compare the values.

44. What makes a HashSet different from a TreeSet?

Answer:

Although both HashSet and TreeSet are not synchronized and ensure that duplicates are not present, there are certain properties that distinguish a HashSet from a TreeSet.

Some of our placed students from Complete Masterclass



Placed in Cognizant GenC
10 LPA



Placed in BMC Software
12.50 LPA



Placed in TCS Digital
7 LPA



- Implementation: For a HashSet, the hash table is utilized for storing the elements in an unordered manner. However, TreeSet makes use of the red-black tree to store the elements in a sorted manner.
- Complexity/ Performance: For adding, retrieving, and deleting elements, the time amortized complexity is $O(1)$ for a HashSet. The time complexity for performing the same operations is a bit higher for TreeSet and is equal to $O(\log n)$. Overall, the performance of HashSet is faster in comparison to TreeSet.
- Methods: hashCode() and equals() are the methods utilized by HashSet for making comparisons between the objects. Conversely, compareTo() and compare() methods are utilized by TreeSet to facilitate object comparisons.
- Objects type: Heterogeneous and null objects can be stored with the help of HashSet. In the case of a TreeSet, runtime exception occurs while inserting heterogeneous objects or null objects

45. Why is the character array preferred over string for storing confidential information?

Answer:

In Java, a string is basically immutable i.e. it cannot be modified. After its declaration, it continues to stay in the string pool as long as it is not removed in the form of garbage. In other words, a string resides in the heap section of the memory for an unregulated and unspecified time interval after string value processing is executed.

As a result, vital information can be stolen for pursuing harmful activities by hackers if a memory dump is illegally accessed by them. Such risks can be eliminated by using mutable objects or structures like character arrays for storing any variable. After the work of the character array variable is done, the variable can be configured to blank at the same instant. Consequently, it helps in saving heap memory and also gives no chance to the hackers to extract vital data.

Complete Placement Preparation LIVE Masterclass

Aptitude | Coding | Certifications & Upskilling | Mock Interviews & Interview Preparation

Click to know more





Join WhatsApp Group for Placement



46. How do exceptions affect the program if it doesn't handle them?

Answer:

Exceptions are runtime errors. Suppose we are making an android application with java. And it all works fine but there is an exceptional case when the application tries to get the file from storage and the file doesn't exist (This is the case of exception in java). And if this case is not handled properly then the application will crash. This will be a bad experience for users. This is the type of error that cannot be controlled by the programmer. But programmers can take some steps to avoid this so that the application won't crash. The proper action can be taken at this step.

47. Although inheritance is a popular OOPs concept, it is less advantageous than composition. Explain.

Answer:

Inheritance lags behind composition in the following scenarios:

- Multiple-inheritance is not possible in Java. Classes can only extend from one superclass. In cases where multiple functionalities are required, for example - to read and write information into the file, the pattern of composition is preferred. The writer, as well as reader functionalities, can be made use of by considering them as the private members.
- Composition assists in attaining high flexibility and prevents breaking of encapsulation.
- Unit testing is possible with composition and not inheritance. When a developer wants to test a class composing a different class, then Mock Object can be created for signifying the composed class to facilitate testing. This technique is not possible with the help of inheritance as the derived class cannot be tested without the help of the superclass in inheritance.
- The loosely coupled nature of composition is preferable over the tightly coupled nature of inheritance.

48. What are Composition and Aggregation? State the difference.

Answer:

Composition, and Aggregation help to build (Has - A - Relationship) between classes and objects. But both are not the same in the end. Let's understand with the help of an example.

Consider the University as a class that has some departments in it. So the university will be the container object. And departments in it will contain objects. Now in this case, if the container object destroys then the contained objects will also get destroyed automatically. So here we can say that there is a strong association between the objects. So this Strong Association is called Composition.

Now consider one more example. Suppose we have a class department and there are several professors' objects there in the department. Now if the department class is destroyed then the professor's object will become free to bind with other objects. Because container objects (Department) only hold the references of contained objects (Professor's). So here is the weak association between the objects. And this weak association is called Aggregation.

49. How is the 'new' operator different from the 'newInstance()' operator in java?

Answer:

Both 'new' and 'newInstance()' operators are used to creating objects. The difference is- that when we already know the class name for which we have to create the object then we use a new operator. But suppose we don't know the class name for which we need to create the object, Or we get the class name from the command line argument, or the database, or the file. Then in that case we use the 'newInstance()' operator.

The 'newInstance()' keyword throws an exception that we need to handle. It is because there are chances that the class definition doesn't exist, and we get the class name from runtime. So it will throw an exception



Follow us on Instagram



50. Is exceeding the memory limit possible in a program despite having a garbage collector?

Answer:

Yes, it is possible for the program to go out of memory in spite of the presence of a garbage collector. Garbage collection assists in recognizing and eliminating those objects which are not required in the program anymore, in order to free up the resources used by them.

In a program, if an object is unreachable, then the execution of garbage collection takes place with respect to that object. If the amount of memory required for creating a new object is not sufficient, then memory is released for those objects which are no longer in the scope with the help of a garbage collector. The memory limit is exceeded for the program when the memory released is not enough for creating new objects.

Moreover, exhaustion of the heap memory takes place if objects are created in such a manner that they remain in the scope and consume memory. The developer should make sure to dereference the object after its work is accomplished. Although the garbage collector endeavors its level best to reclaim memory as much as possible, memory limits can still be exceeded.

**Talent Battle is
associated with TCS iON**

for content partnership &
providing internships to
students across India.





Join WhatsApp Group for Placement



51. Why is synchronization necessary? Explain with the help of a relevant example.

Answer:

Concurrent execution of different processes is made possible by synchronization. When a particular resource is shared between many threads, situations may arise in which multiple threads require the same shared resource.

Synchronization assists in resolving the issue and the resource is shared by a single thread at a time. Let's take an example to understand it more clearly. For example, you have a URL and you have to find out the number of requests made to it. Two simultaneous requests can make the count erratic.

52. What could be the tradeoff between the usage of an unordered array versus the usage of an ordered array?

Answer:

- The main advantage of having an ordered array is the reduced search time complexity of $O(\log n)$ whereas the time complexity in an unordered array is $O(n)$.
- The main drawback of the ordered array is its increased insertion time which is $O(n)$ due to the fact that its element has to reordered to maintain the order of array during every insertion whereas the time complexity in the unordered array is only $O(1)$.
- Considering the above 2 key points and depending on what kind of scenario a developer requires, the appropriate data structure can be used for implementation.

53. What is the best way to inject dependency? Also, state the reason.

Answer:

- There is no boundation for using a particular dependency injection. But the recommended approach is -
- Setters are mostly recommended for optional dependencies injection, and constructor arguments are recommended for mandatory ones. This is because constructor injection enables the injection of values into immutable fields and enables reading them more easily.



Get free mentorship
from experts



54. How we can set the spring bean scope. And what supported scopes does it have?

Answer:

A scope can be set by an annotation such as the @Scope annotation or the "scope" attribute in an XML configuration file. Spring Bean supports the following five scopes:

- Singleton
- Prototype
- Request
- Session
- Global-session

Free Pre - Placement Mock Test Series - Aptitude & Technical



Why we are conducting Pre-Placement Test Series?

- To help students check their current level of Aptitude and Technical skills
- After knowing the current level it will become easy for a student to start their placement preparation

What does this Pre-Placement Test Series consist of?

- 25 Tests will be conducted on subjects C, C++, Java, Python, DSA, CN, OS, DBMS, Quant, Reasoning, and verbal ability.
- Topic-wise questions in every test will help students get strong and weak points

Introducing...

Complete Placement

Preparatory Masterclass



What is included?



400+ Hours Foundation of LIVE + Recorded Training on **Quant, Reasoning, Verbal, C, C++, Java, Python, DSA, OS, CN, DBMS**



Interview preparation Training along with One-to-One Mock **Technical & Personal Interviews** by experts



Latest Technologies Certification Courses to get higher packages. 500+ hours of courses on **Full stack development, AI, ML, Data science, Data Analytics, Tableau, PowerBI and much more**



Company-specific LIVE and Recorded training for **30+ service and product-based companies.**

TCS NQT | Accenture | Capgemini | Cognizant | Infosys | Persistent | Deloitte | Mindtree | Virtusa | Goldman Sachs | Bosch | Samsung Amazon | Nalsoft | Zoho Cisco and 15+ more companies preparation.



15+ Real Time Projects based on Latest Technologies and **10+ Mini Projects** based on C, C++, Java and Python to build your Profile



Get TCS iON Internship / TCS NQT paid exam for free
Certificate of Internship from TCS iON

Our Placement Reports

97.6% 

Selection Ratio
of Complete Masterclass Students

4.91 / 5



Overall Rating
out of 5

Highest CTC
40 LPA

85% students
placed with CTC more
than 6 LPA

Average CTC
8.5 LPA

Referral Opportunities
in Top Companies & Startups
by Talent Battle

Average Offers
per student: 2.7

Students from
5000+ colleges
use Masterclass for
Placement Preparation

- **50000+ placed** students (in the last 10 years)
- Our Students placed in **600+ Companies**
- **10 Lakh+ Students** Prepare with Talent Battle Resources Every Year

Talent Battle Certifications



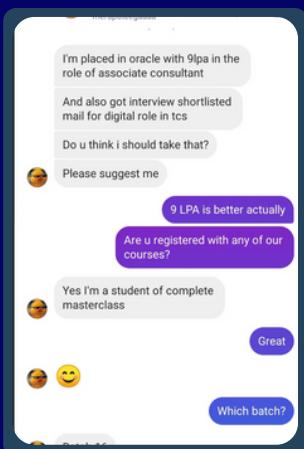
Get a Free Mentorship from experts for your Campus Placement Preparation

- Discuss your queries with experts
- Get a roadmap for your placement preparation

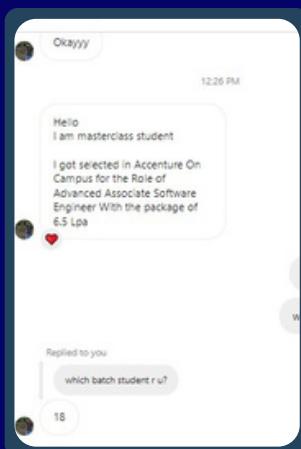
Click to know more



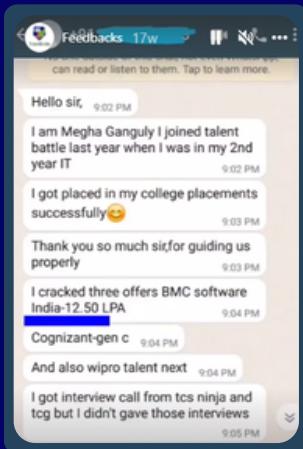
Some of our placed students



Srinija Kalluri
COMPLETE MASTERCLASS STUDENT
SELECTED AT ORACLE - 9 LPA



Rohit
COMPLETE MASTERCLASS STUDENT
SELECTED AT ACCENTURE - 6.5 LPA



Megha Ganguly
COMPLETE MASTERCLASS STUDENT
SELECTED COGNIZANT, WIPRO & BMC INDIA - 12.5 LPA



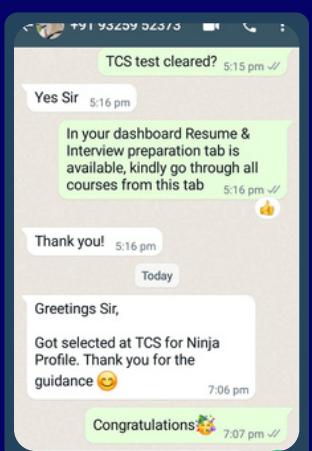
Aditya Kulesetha
COMPLETE MASTERCLASS STUDENT
SELECTED AT COGNIZANT - 7.8 LPA



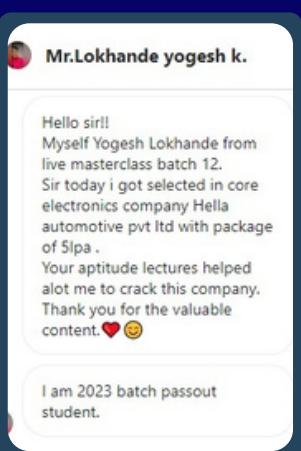
Shubham Verma
COMPLETE MASTERCLASS STUDENT
SELECTED AT CAPGEMINI - 7.5 LPA



Amardeep Prajapati
COMPLETE MASTERCLASS STUDENT
SELECTED AT HAPPIEST MIND TECHNOLOGY - 5.4 LPA



Rutuja Jangam
COMPLETE MASTERCLASS STUDENT
SELECTED AT TCS NINJA



Yogesh Lokhande
COMPLETE MASTERCLASS STUDENT
SELECTED AT HELLA ELECTRONICS -5 LPA



Vikas Varak
COMPLETE MASTERCLASS STUDENT
SELECTED AT TCS NINJA

Tools & Technologies

covered in Placement Pro!



Our Team



Amit Prabhu

Co-founder

9+ years of experience in training students for Quantitative Aptitude, Verbal & Monitoring students for Campus Placement Preparation



Ajinkya Kulkarni

Co-founder

9+ years of experience in training students for Reasoning, Interview Training & Mentoring Students for Campus Placement Preparation



Rohit Bag

Lead Technical Trainer

10+ years of experience in training students for Programming Languages & Core Computer Science Subjects along with Company Specific Training



Vaishnavi K Dharan

Technical Trainer

5+ years of experience in training students on different Programming Languages and Data Structure. Master certified trainer in Robotic Process Automation in Automation Anywhere.



Poojitha Renati

Lead Aptitude Trainer

5+ years of experience in training students for Aptitude and Logical Reasoning. Trained more than 5000+ hours in various institutes including GITAM, Parul, KITS, JNTU and more



Chand Basha

Lead Aptitude Trainer

8+ years of experience in training students Quantitative Aptitude, Logical Reasoning & Verbal Ability for Campus Placements & Competitive Exams



Jasleen Chhabra
Mentor-Training and Placement

3+ years of experience in dealing with students and their counselling related to Academic problems as well as their placements.



Samradhni Wankhede
Graphic Designer

4+ years experience as a Creativity Expert, Graphic Designer, Teacher/Trainer and a social media enthusiast



Akshay Paricharak
Customer Service Manager

8+ years of experience in Customer service, Students counselling, Business development, Project co-ordination, Strategies Implementation, Planning and Execution.



Niranjan Kulkarni
Program Manager - Training and Placement

15 years of overall multi-functional experience in Industry and Academia, in managing diverse functions such as Training, and Human Resource Management.



Ruturaj Sankpal
Placement Mentor

2 years of experience in IT industry and currently mentoring students for campus placements.



Swapnil Wankhede
Marketing & Ops.

2.5+ years of experience in compassionate and ethical marketing. Striving to ensure students get the best opportunities and are prepared to make the most of it, learning and growing with Talent Battle.

Industry Mentors



Sandip Karekar
Industry Mentor

8 years of Industry experience and currently working with Mastercard. Decent understanding of core technologies in Computer Science & Information Technology and passionate about learning Cutting-Edge technologies.



Mayuresh Diwan
Industry Mentor

Placement Mentor: 4+ years of experience in automotive software development.



Swapnil Patil
Industry Mentor

Lead Engineer at John Deere
Over 4+ years of experience as a Software Developer. Having expertise in developing and maintaining web and desktop applications of different domains like Telecom, Simulations and Automations, ERP and CRM



Shadab Gada
Industry Mentor

Software developer with 3+ years of experience in design, developing, testing and maintenance of web based applications. Passionate about learning new technologies, always eager to share my knowledge, love interacting with new people.

FAQs

1 What is Talent Battle?

Talent battle is a group of mentors and educators who help students to prepare for their On and Off campus placement opportunities. We have trainers with an average of 10 years of experience in training students for their Tech company drives. We train students on Aptitude, Programming, communication skills, projects, advance technologies and all other necessary skills to get placed in their dream companies. If you want to get placed in any of your dream companies, then join our complete masterclass and fulfill your dream!

2 When and how to prepare for campus placements?

The best time to start preparing for your campus is in your third year of engineering. During this time you can start preparing for your Aptitude, Verbal and Programming skills.

Most of the companies have a similar testing pattern for selecting students. There are typically tests on aptitude, programming and communication skills. The short answer for this question is, **prepare with Talent Battle's Masterclass** as we cover all of the above mentioned topics in detail.

3 What is Complete Masterclass?

Complete Masterclass is a combination of Concept Clearing lectures for Aptitude, Coding, DSA + Company Specific Training for 30+ Companies + Interview Preparation with Mock Interviews. Foundational and Company Specific Training is conducted LIVE. Whenever companies launch their drives, we will be conducting company specific live sessions. Foundational training right from basic to advance level will also be available in recorded format.

Along with that we have 240+ hours of Full stack Development course, either of TCS ion internship or PAID NQT (for 2 years package) and 250+ hours of Advance certification courses like AI, ML, Data Science, etc will be available free of cost.

4 Why to chose Talent Battle?

We have structured and disciplined way of preparation. You don't need to seek outside information source. All the study material will be available on Talent Battle's dashboard. We provide end to end support to our students until they get placed. Talent Battle is one stop solution to prepare for placement drives.

Dont delay your placement preparation anymore!!

Learn from the experts!

Check out our social media to get regular placement related content & job drive updates

-  @talentbattle.in
-  @talentbattle_2023
-  @talentbattle_2024
-  @talentbattle_2025
-  @talentbattle_2026
-  WhatsApp Group
-  Free Mentorship
-  Talent Battle Facebook
-  Talent Battle YouTube
-  Talent Battle LinkedIn
-  <https://talentbattle.in/>

