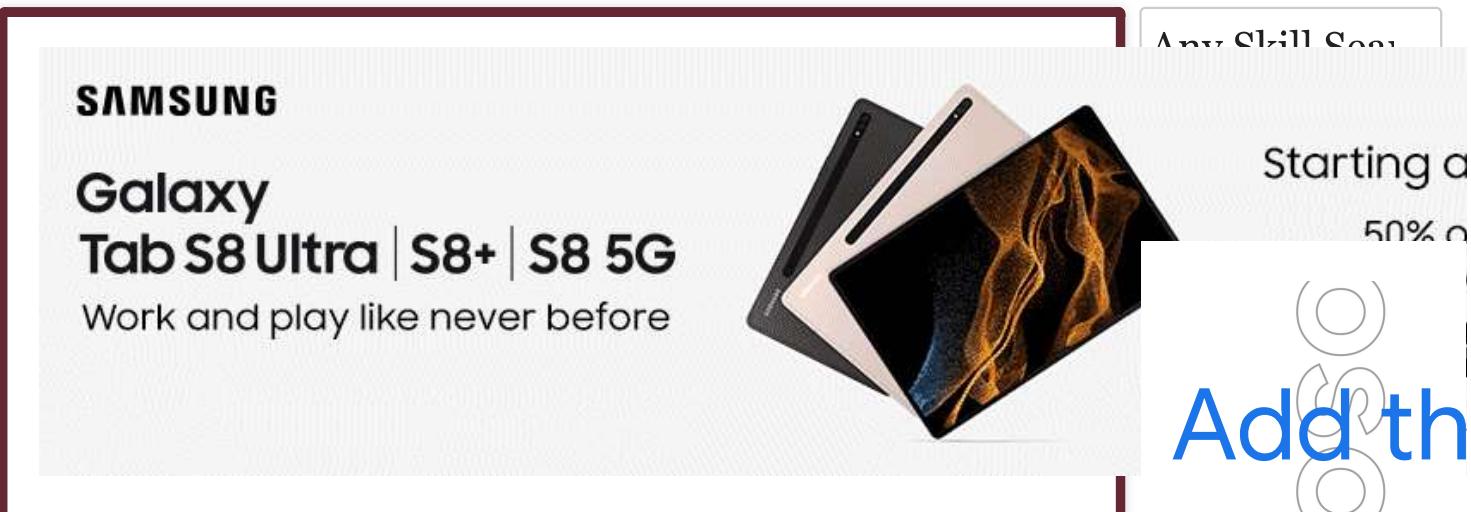


# Interview Questions | MCQs

HOME Interview Questions MCQs Class Notes \*LAB VIVA SEMINAR TOPICS  
ONLINE TEST GATE CAT Internship ABOUT US



A banner advertisement for Samsung Galaxy tablets. It features three tablets: a black S8 Ultra, a white S8+, and a dark S8 5G. The S8 Ultra screen displays a vibrant orange and blue abstract pattern. The text "SAMSUNG" is at the top left, followed by "Galaxy Tab S8 Ultra | S8+ | S8 5G" and "Work and play like never before". To the right, there's a small snippet of another page with text "Starting a" and "50% o" and some icons.

Any Skill Set

Starting a

50% o



Add th  
extens



Microsoft

[Home](#) » [Java Inner Classes Questions](#) » **300+ [LATEST] Java Inner Classes Interview Questions and Answers**

## 300+ [LATEST] Java Inner Classes Interview Questions and Answers

Build a Website wit

Learn to create cross-platf

Download

Scaler Academy

Q1. What Is Static Member Class?



A static member class behaves much like an ordinary top-level class, except that it can access the static members of the class that contains it. The static nested class can be accessed as the other static members of the enclosing class without having an instance of the outer class. The static class can contain non-static and static members and methods.

```
public class InnerClass
```

```
{
```

```
    static class StaticInner
```

```
{
```

```
    static int i = 9;
```

```
    int no = 6;
```

```
    private void method() {}
```

```
    public void method1() {}
```

```
    static void method2() {}
```

```
    final void method3() {}
```

```
}
```

```
}
```

**The static inner class can be accessed from Outer Class in the following manner:**

- `InnerClass.StaticInner staticObj= new InnerClass. StaticInner();`
- No outer class instance is required to instantiate the nested static class because the static class is a static member of the



enclosing class.

## Q2. What Is An Inner Class?

Inner class is a class defined inside another class and act like a member of the enclosing class.

## Q3. What Are Disadvantages Of Using Inner Classes?

- Using inner class increases the total number of classes being used by the application. For all the classes created by JVM and loaded in the memory, jvm has to perform some tasks like creating the object of type class. Jvm may have to perform some routine tasks for these extra classes created which may result slower performance if the application is using more number of inner classes.
- Inner classes get limited support of ide/tools as compared to the top level classes, so working with the inner classes is sometimes annoying for the developer.

## Q4. Can An Anonymous Class Define Method Of Its Own?

Yes. But there will be no way by which the methods defined in the anonymous class which are not present in its superclass be invoked. As only those methods which are defined in the superclass which the anonymous class extends be invoked. Hence defining the methods in the anonymous class will be of no use.



# Complete Test Series

for Product-Based Companies

**Practice 1000+**

Company Specific Interview Questions

[Register Now](#)



Q5. Can The Method Local Inner Class Object Access Method Local Variables?

# Complete Test Series

for Product-Based Companies

**Practice 1000+**

Company Specific Interview Questions

[Register Now](#)



No, a method local inner class object cannot access the method local variable.

**Reason:** The local variables are not guaranteed to live as long as the local inner class objects. The method local variable live on stack and exist only till the method lives, their scope is limited upto the code inside the method they are declared in. But the local inner class object created within the method lives on heap and it may exist even after



the method ends if in case the reference of this local inner class is passed into some other code and is stored in an instance variable. So we cannot be sure that the local variables will live till the method local inner class object lives, therefore the method local inner class object cannot access the method local variable. To access the method local variables, the variable has to be declared as final.

#### Q6. Can A Method Local Inner Class Access The Local Final Variables? Why?

Yes. Because the final variables are stored on heap and live as long as the method local inner class objects may live.

#### Q7. Which Modifiers Can Be Applied To The Inner Class?

##### **Following modifiers can be applied to the inner class:**

- Public
- Private
- Abstract
- Final
- protected
- Strict
- Static – turns the inner class into
- Static nested class

#### Q8. If You Compile A File Containing Inner Class How Many .class Files Are Created And Are All Of Them Accessible In Usual Way?

If a inner class enclosed with an outer class is compiled then one .class file for each inner class and a .class file for the outer class is created.

##### **Example:**

```
class EnclosingOuter
```

```
{
```



```
class Inner{ }  
}
```

**If you compile the above code with command:**

% javac EnclosingOuter.java Two files will be created. Though a separate inner class file is generated, the inner class file is not accessible in the usual way.

EnclosingOuter.class

EnclosingOuter\$Inner.class

**Q9. What Are The Different Types Of Inner Classes?**

**The below mentioned are the types of inner classes:-**

- Static member class
- Inner class
- Member class
- Anonymous class
- Local class

**Q10. How To Access The Inner Class From Code Within The Outer Class?**

The inner class is instantiated only through the outer class instance.

```
class EnclosingOuter  
{  
    private int noInnerClass = 1;  
  
    public void getNoOfInnerClasses()  
    {
```



```
Inner in = new Inner();  
  
System.out.println("No Of Inner classes is :" +  
in.getNumberOfClassesFromOuter());  
  
}  
  
class Inner  
  
{  
  
public int getNumberOfClassesFromOuter()  
  
{ return noInnerClass;  
  
}  
  
}
```

Here the method `getNumberOfInnerClasses()` is called on the outer class's instance and through this outer class instance the inner class instance is created.

**Q11. Can A Local Class Declared Inside A Static Method Have Access To The Instance Members Of The Outer Class?**



# Complete Test Series

for Product-Based Companies

**Practice 1000+**  
Company Specific Interview Questions

[Register Now](#)

**GG** GeeksforGeeks

No. There is no reference available in the static method about the instance members of the outer class. The static method class cannot have access to any members of the outer class other than static members.

## Q12. What Are Non Static Inner Classes?

**The different type of static inner classes is:** Non – static inner classes – classes associated with the object of the enclosing class.

**Member class** – Classes declared outside a function (hence a “member”) and not declared “static”. The member class can be declared as public, private, protected, final and abstract.

### Example:

```
public class InnerClass
```

```
{
```

```
class MemberClass
```

```
{
```



```
public void method1() { }  
}  
}
```

**Method local class** – The inner class declared inside the method is called method local inner class. Method local inner class can only be declared as final or abstract. Method local class can only access global variables or method local variables if declared as final

```
public class InnerClass
```

```
{
```

```
int i = 9;
```

```
public void method1()
```

```
{
```

```
final int k = 6;
```

```
class MethodLocal
```

```
{
```

```
MethodLocal()
```

```
{
```

```
System.out.println(k + i);
```

```
}
```

```
}
```

```
}
```



}

**Anonymous inner class** – These are local classes which are automatically declared and instantiated in the middle of an expression. Also, like local classes, anonymous classes cannot be public, private, protected, or static. They can specify arguments to the constructor of the superclass, but cannot otherwise have a constructor. They can implement only one interface or extend a class.

```
public class MyFrame extends JFrame
```

{

```
 JButton btn = new JButton();
```

```
MyFrame()
```

{

```
btn.addActionListener(new ActionListener()
```

{

```
public void actionPerformed(ActionEvent e) {
```

});

}

}

```
//Anonymous class used with comparator
```

```
List l = new ArrayList();
```

```
l.add(new Parent(2));
```

```
l.add(new Parent(3));
```



```
Collections.sort(l, new Comparator()
```

```
{
```

```
    public int compare(Object o1, Object o2)
```

```
{
```

```
    Parent prt1 = (Parent) o1;
```

```
    Parent prt2 = (Parent) o2;
```

```
    if (prt1.getAge() > prt2.getAge())
```

```
{
```

```
    return -1;
```

```
}
```

```
    else if(prt1.getAge() {
```

```
        return 1;
```

```
}
```

```
    else
```

```
{
```

```
    return 0;
```

```
}
```

```
}
```

```
});
```



### Q13. Which Modifiers Can Be Applied To The Method Local Inner Class?

Only abstract or final keyword is allowed.

### Q14. Can An Anonymous Class Implement An Interface And Also Extend A Class At The Same Time?

No. An anonymous class can either extend a class or implement a single interface. If the anonymous class is extending a class then it automatically becomes the implementer of all the interfaces implemented by its superclass.

### Q15. What Are The Advantages Of Inner Classes?

The embedding of inner class into the outer class in the case when the inner class is to be used only by one class i.e. the outer class makes the package more streamlined. Nesting the inner class code where it is used (inside the outer class) makes the code more readable and maintainable. The inner class shares a special relationship with the outer class i.e.

the inner class has access to all members of the outer class and still have its own type is the main advantages of Inner class. Advantage of inner class is that they can be hidden from the other classes in the same package and still have the access to all the members (private also) of the enclosing class. So the outer class members which are going to be used by the inner class can be made private and the inner class members can be hidden from the classes in the same package.



This increases the level of encapsulation. If a class A is written requires another class B for its own use, there are two ways to do this. One way is to write a separate class B or to write an inner class B inside class A. Advantage of writing the inner class B in the class A is you can avoid having a separate class. Inner classes are best used in the event handling mechanism and to implement the helper classes. The advantage of using inner class for event handling mechanism is that the use of if/else to select the component to be handled can be avoided. If inner classes are used each component gets its own event handler and each event handler implicitly knows the component it is working for.

#### Q16. What Are Different Types Of Anonymous Classes?

##### **Plain old anonymous class type one:-**

```
class superclass{
```

```
void doSomething() {
```

```
System.out.println("Doing something in the Super class");
```

```
}
```

```
}
```

```
class hasAnonymous{
```



```
superClass anon = new superClass(){  
  
void doSomething() {  
  
System.out.println("Doing something in the Anonymous class");  
  
}  
  
};
```

Here anon is the reference which is of type superclass which is the class extended by the anonymous class i.e. superclass of the anonymous class. The method doSomething () is the super class method overridden by the anonymous class.

### **Plain old anonymous class type two:-**

```
interface Eatable {  
  
public void prepareSweets();  
  
}  
  
class serveMeal {  
  
Eatable food = new Eatable(){  
  
public void  
  
prepareSweets(){ //come implementation code goes here  
  
}  
  
};  
  
}
```



food is reference variable of type Eatable interface which refers to the anonymous class which is the implementer of the interface Eatable. The anonymous implementer class of the interface Eatable implements its method prepare Sweets () inside it.

### Argument defined anonymous class:-

```
interface Vehicle {  
    void getNoOfWheels();  
}  
  
class Car {  
    void getType(Vehical v) {}  
}  
  
class BeautifulCars  
{  
    void getTheBeautifilCar()  
    {  
        Car c = new Car ();  
        c.getType (new Vehicle ()  
        {  
            public void getNoOfWheels ()  
            {  
                System.out.println("It has four wheels");  
            }  
        });  
    }  
}
```



```
}
```

```
});
```

```
}
```

```
}
```

Anonymous class is defined as the argument of the method `getTheBeautifulCar ()`, this anonymous class is the implementer of the interface `Vehicle`. The method of class `Car` `getTheBeautifulCar ()` expects the argument as an object of type `Vehicle`. So first we create an object of `Car` referenced by the variable '`c`'. On this object of `Car` we call the method `getTheBeautifulCar ()` and in the argument we create an anonymous class in place which is the implementer of interface `Vehicle` hence of type `Vehicle`.

**Q17. How To Create An Inner Class Instance From Outside The Outer Class Instance Code?**

To create an instance of the inner class you must have the instance of its enclosing class.

```
class Enclosing Outer
```

```
{
```



```
class Inner{ }
```

```
}
```

To create the instance of inner class from class other than the enclosing class.

1) class OtherThanOuter

```
{
```

```
Enclosing Outer out = new Enclosing Outer();
```

```
EnclosingOuter.Inner in = out.new Inner();
```

```
}
```

2) class OtherThanOuter

```
{
```

```
EnclosingOuter.Inner out = new EnclosingOuter.Inner();
```

```
}
```

Q18. Can A Method Which Is Not In The Definition Of The Superclass Of An Anonymous Class Be Invoked On That Anonymous Class Reference?

No. As the reference variable type of the anonymous class will be of superclass which will not know about any method defined inside the anonymous class and hence the compilation will fail.

```
class Superclass
```

```
{
```

```
void doSomething()
```



```
{  
    System.out.println ("In the Super class");  
}
```

```
}  
}  
  
class has Anonymous
```

```
{  
    Superclass anon = new Superclass ()  
{
```

```
    void doSomething()  
{
```

```
    System.out.println("In the Anonymous class");  
}
```

```
void doStuff()
```

```
{  
    System.out.println ("An Anonymous class method not present in  
superclass");
```

```
}  
};
```

```
public void doIt()
```



```
{  
anon. DoSomething(); // legal superclass has this method  
  
anon. DoStuff(); // Not legal  
  
}  
  
}
```

The above code will not compile as the superclass does not know about the anonymous class method doStuff () .

**Q19. Can A Static Nested Class Have Access To The Enclosing Class's Non-static Methods Or Instance Variables?**

No.

**---- >> Below Are The Related Posts Of Above Questions :::**

**----->>[MOST IMPORTANT]<<-----**

- 1. [300+ \[LATEST\] Java Abstraction Interview Questions and Answers](#)**
- 2. [300+ \[LATEST\] Java Programmer Interview Questions and Answers](#)**
- 3. [300+ \[LATEST\] Java Developer Interview Questions and Answers](#)**
- 4. [300+ \[LATEST\] Java Generics Interview Questions and Answers](#)**
- 5. [300+ \[LATEST\] Java Programming With Ooops Concepts Interview Questions and Answers](#)**
- 6. [300+ \[LATEST\] Java Webdynpro Interview Questions and Answers](#)**
- 7. [300+ \[LATEST\] Java.lang Interview Questions and Answers](#)**



- 8. 300+ [LATEST] Java.util Interview Questions and Answers**
- 9. 250+ TOP MCQs on Nested Class and Answers**
- 10. 300+ [LATEST] Java Exception Handling Interview Questions and Answers**
- 11. 300+ [LATEST] Java Hadoop Developer Interview Questions and Answers**
- 12. 300+ [LATEST] Aricent Java Interview Questions and Answers**
- 13. 300+ [LATEST] Ericsson Java Interview Questions and Answers**
- 14. 300+ [LATEST] Java Inheritance Interview Questions and Answers**
- 15. 250+ TOP MCQs on Classes and Answers**
- 16. 300+ [LATEST] Java Serialization Interview Questions and Answers**
- 17. 300+ [LATEST] Java.lang Package Interview Questions and Answers**
- 18. 300+ [LATEST] Java-multithreading Interview Questions and Answers**
- 19. 300+ [MOSK ASKED] Birlasoft Java Interview Questions and Answers**
- 20. 300+ [LATEST] Java Garbage Collection Interview Questions and Answers**

Engineering 2022 , FAQs Interview Questions , Theme by Engineering|| Privacy Policy|| Terms and Conditions|| ABOUT US|| Contact US||

Engineering interview questions,Mcqs,Objective Questions,Class Lecture Notes,Seminor topics,Lab Viva Pdf PPT Doc Book free download. Most Asked Technical Basic CIVIL | Mechanical | CSE | EEE | ECE | IT | Chemical | Medical MBBS Jobs Online Quiz Tests for Freshers Experienced .

