[**Java Interview Questions On Modifiers**](http://javaconceptoftheday.com/java-interview-questions-on-modifiers/)

1. **How many types of modifiers are there in Java?**

Two types of modifiers are there in java. They are,

a) Access Modifiers

b) Non-access Modifiers

**2) What are access modifiers in java.?**

These are the modifiers which are used to restrict the visibility of a class or a field or a method or a constructor. Java supports 4 access modifiers.

**a) private** : private fields or methods or constructors are visible within the class in which they are defined.

**b) protected** : Protected members of a class are visible within the package but they can be inherited to sub classes outside the package.

**c) public :** public members are visible everywhere.

**d) default or No-access modifiers :** Members of a class which are defined with no access modifiers are visible within the package in which they are defined.

**3) What are non-access modifiers in java?**

These are the modifiers which are used to achieve other functionalities like,

**a) Static:** This modifier is used to specify whether a member is a class member or an instance member.

**b) Final:** It is used to restrict the further modification of a class or a method or a field.

**c) Abstract:** abstract class or abstract method must be enhanced or modified further.

**d) Synchronized:** It is used to achieve thread safeness. Only one thread can execute a method or a block which is declared as synchronized at any given time.

**4) Can we use a field or a method declared without access modifiers outside the package?**

No, we can’t use a field or a method with no-access (default) Specifiers outside the package in which their class is defined.

**5) Can a method or a class be final and abstract at the same time?**

No, it is not possible. A class or a method cannot be final and abstract at the same time. Final and abstract are totally opposite in nature. Final class or final method must not be modified further whereas abstract class or abstract method must be modified further.

**6) Can we declare a class as private?**  
 Yes… we can declare a class as a private… but it is must be a inner class of the class…. for example

**class A {**

**private class B{**

**}**

**}**

Here class B can accessible with in class A… successfully

**7) Can we declare an abstract method as private also?**

No, abstract methods cannot be private. They must be public or protected or default so that they can be modified further.

**8) Can we declare a class as protected?**

We can’t declare an outer class as protected. But, we can declare an inner class (class as a member of another class) as protected.

**9) A class cannot be declared with synchronized keyword. Then, why we call classes like Vector, StringBuffer are synchronized classes?**

Any classes which have only synchronized methods and blocks are treated as synchronized classes. Classes like Vector, StringBuffer have only synchronized methods. That’s why they are called as synchronized classes.

**What is the use of access modifiers in Java?**

Access Modifiers in Java are used to control the visibility of fields, methods, classes and constructors.

**Can you create a sub class to the following class?**

|  |  |
| --- | --- |
|  | ***class A{***  ***private A()    {***  ***//First Constructor***  ***}***  ***private A(int i)    {***  ***//Second Constructor***  ***}***  ***}*** |

No, you can’t create sub classes to that class which has only private constructors.

**3) Can you find out the error in the below code?**

|  |  |
| --- | --- |
|  | **private class A{**  **private class B    {**  **//Inner class**  **}**  **}** |

Inner classes can be private, but outer classes cannot be private.

**4) Does field ‘i’ of Class A be inherited to Class B in the below code?**

|  |  |
| --- | --- |
|  | **class A{**  **protected int i;**  **}**    **class B extends A{**  **}** |

Yes, protected members of a class are inherited to any sub class.

**5) Is the below code written correctly?**

|  |  |
| --- | --- |
|  | **class A{**  **private class B    {**  **//inner class**  **}**  **}**    **public class MainClass extends A{**  **public static void main(String[] args) {**  **B b = new B();**  **}**  **}** |

No. private inner Class B cannot be instantiated outside the Class A.

**6) Is the below code written correctly?**

|  |
| --- |
| **package pack1;**  **class A{**  **}**  **package pack2;**  **class B extends A{**  **}** |

No. Class with default (no) access modifiers cannot have subclass outside the package.

**7) Can we declare a class as protected?**

Yes, but only inner class. Outer class cannot be protected.

**8) Do you think the below program is written correctly? If yes, what will be the output?**

|  |  |
| --- | --- |
|  |  |

Yes, it is written correctly. Output will be  
1221  
1221

**9) Why we can’t instantiate Class-A in the below code outside the package even though it has public constructor?**

|  |  |
| --- | --- |
|  | **package pack1;**  **class A{**  **public A()    {**  **//public constructor**  **}**  **}**    **package pack2;**  **import pack1.\*;**  **class B{**  **A a = new A();       //Compile Time Error**  **}** |

**Answer**  
Because, Class-A itself has been defined with default access modifier. That means Class-A can be instantiated within the package in which it is defined. It cannot be instantiated outside the package, even though it has public constructor.

**10) Can a protected field of a class be inherited to subclass outside the package?**

**Answer**  
Yes protected members of a class are inherited to sub classes outside the package.

**11) Why Line 17 in the below code is throwing compile time error?**

|  |  |
| --- | --- |
|  | ***package pack1;***  ***public class A{***  ***protected A()    {***  ***//protected constructor***  ***}***  ***}***  ***package pack2;***  ***import pack1.A;***  ***class B{***  ***A a = new A();***  ***}*** |

**Answer**  
because, we can’t instantiate a class outside the package which has only protected constructors.

**12) Do you think the below code compiles successfully even though it is calling super class’s protected constructor outside the package?**

|  |  |
| --- | --- |
|  | **package pack1;**  **public class A{**  **protected A(int i)    {**  **//protected constructor**  **}**  **}**    **package pack2;**  **import pack1.A;**  **class B extends A{**  **public B()    {**  **super(10);     //calling super class's protected constructor**  **}**  **}** |

**Answer**  
yes, above code will compile successfully.

**13) Can we declare static methods as private?**

**Answer**  
Yes, static members of a class can be private.

**14) Is the below code written correctly? If yes, what will be the output?**

|  |  |
| --- | --- |
|  | **package pack1;**  **class A{**  **protected static String s = "A";**  **}**  **class B extends A{**  **}**    **class C extends B{**  **static void methodOfC()    {**  **System.out.println(s);**  **}**  **}**    **public class MainClass{**  **public static void main(String[] args)    {**  **C.methodOfC();**  **}**  **}** |

**Answer**  
Yes, it is written correctly. Output will be A.

**15) Write the access modifiers in the increasing order of their visibility?**

**Answer**  
private —> default or no access modifiers —> protected —> public

**16) How many public classes a .java file can have?**

**Answer**  
only one. A .java file can have maximum one public class.

**17) What will be the outcome of the below program?**

|  |  |
| --- | --- |
|  | **package pack1;**  **public class A{**  **private int methodOne(int i)    {**  **return ++i;**  **}**  **public int methodTwo(int i)   {**  **return methodOne(++i);**  **}**  **}**  **package pack2;**  **import pack1.A;**  **class B extends A{**  **int methodOne(int i)    {**  **return methodTwo(++i);**  **}**  **}**  **public class MainClass{**  **public static void main(String[] args)    {**  **System.out.println(new B().methodOne(101));**  **}**  **}** |

**Answer**  
104

**18) Can you find out the error in the following code snippet?**

|  |  |
| --- | --- |
|  | **class A{**  **public void methodOfA()    {**  **System.out.println("Class A");**  **}**  **}**    **class B extends A{**  **@Override**  **void methodOfA()    {**  **System.out.println("Class B");**  **}**  **}** |

**Answer**  
The visibility of methodOfA() has been reduced to default while overriding it in the class B. You can’t reduce the visibility of a method while overriding it.

**19) Private method can be overridden as public method. True or False?**

**Answer**  
False. Private methods are not at all inherited.

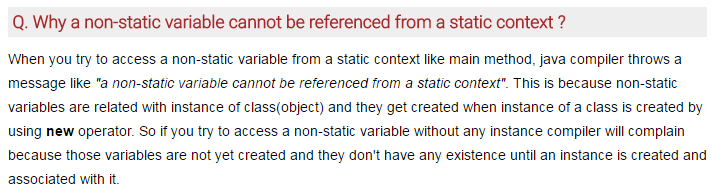
**20) A method of super class with default access modifier can be overridden as protected or public but not as private. True or false?**

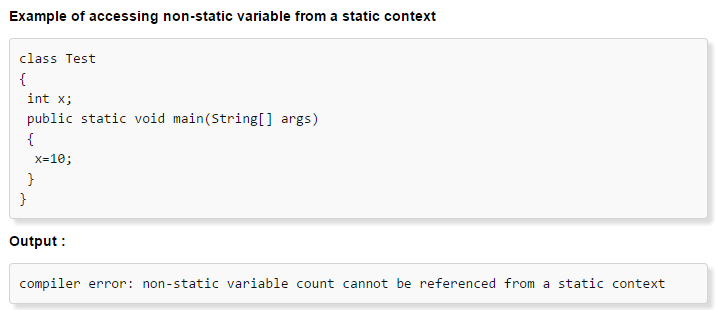
**Answer**   
True.

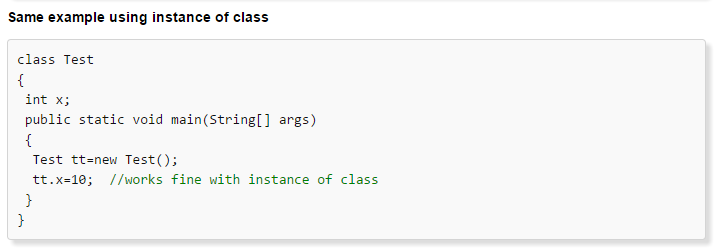
**21) Mona has written the codes like below but it is showing compile time error. You help him to remove the error?**

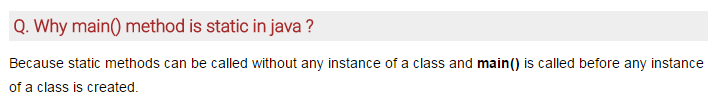
|  |  |
| --- | --- |
|  | **private class A{**  **private class B    {**  **private class C       {**    **}**  **}**  **}** |

**Answer**  
Outer class can’t be private. Don’t declare Class A as private.









**Java Modifiers Important Questions**

1) For the Top Level Classes which Modifiers are allowed?

2) Is it Possible to Declare a Class as static, private, and protected?

3) What is Extra Modifiers Applicable for Inner Classes when compared with Outer Classes?

4) What is a final Class?

5) Explain the Differences between final, finally and finalize?

6) Is Every Method Present in final Class is final?

7) Is Every Variable Present Inside a final Class is final?

8) What is abstract Class?

9) What is abstract Method?

10) If a Class contain at least one abstract Method is it required to declared that Class Compulsory abstract?

11) If a Class doesn’t contain any abstract Methods is it Possible to Declare that Class as abstract?

12) Whenever we are extending abstract Class is it Compulsory required to Provide Implementation for Every abstract Method of that Class?

13) Is final Class can contain abstract Method?

14) Is abstract Class can contain final Methods?

15) Can You give Example for abstract Class which doesn’t contain any abstract

Method?

16) Which of the following Modifiers Combinations are Legal for Methods?

 public – static

 static – abstract

 abstract - final

 final - synchronized

 synchronized - native

 native – abstract

17) Which of the following Modifiers Combinations are Legal for Classes?

 public - final

 final - abstract

 abstract - strictfp

 strictfp – public

18) What is the Difference between abstract Class and Interface?

19) What is strictfp Modifier?

20) Is it Possible to Declare a Variable with strictfp?

21) abstract - strictfp Combination, is Legal for Classes OR Methods?

22) Is it Possible to Override a native Method?

23) What is the Difference between Instance and Static Variable?

24) What is the Difference between General Static Variable and final Static Variable?

25) Which Modifiers are Applicable for Local Variable?

26) When the Static Variables will be Created?

27) What are Various Memory Locations of Instance Variables, Local Variables and Static Variables?

28) Is it Possible to Overload a main()?

29) Is it Possible to Override Static Methods?

30) What is native Key Word and where it is Applicable?

31) What is the Main Advantage of the native Key Word?

32) If we are using native Modifier how we can Maintain Platform Independent Nature?

33) How we can Declare a native Method?

34) Is abstract Method can contain Body?

35) What is synchronized Key Word where we can Apply?

36) What are Advantages and Disadvantages of synchronized Key Word?

37) Which Modifiers are the Most Dangerous in Java?

38) What is Serialization and Explain how its Process?

39) What is Deserialization?

40) By using which Classes we can Achieve Serialization and Deserialization?

41) What is Serializable interface and Explain its Methods?

42) What is a Marker Interface and give an Example?

43) Without having any Method in Serializable Interface, how we can get Serializable Ability for Our Object?

44) What is the Purpose of transient Key Word and Explain its Advantages?

45) Is it Possible to Serialize Every Java Object?

46) Is it Possible to Declare a Method, a Class with transient?

47) If we Declare Static Variable with transient is there any Impact?

48) What is the Impact of declaring final Variable a transient?

49) What is volatile Variable?

50) Is it Possible to Declare a Class OR a Method with volatile?

51) What is the Advantage and Disadvantage of volatile Modifier?