

Java Interview Question and Answers

➤ Topic – Java Static Keyword :

Important Interview Questions based on Static Keyword

1. What is static in Java?

Ans: In Java, static is a keyword that is used for memory management mainly. Static means single copy storage for variables or methods.

The members that are marked with the static keyword inside a class are called static members.

2. Can we access static members if no instance of the class is constructed?

Ans: Yes, we can access the static members if no instance of class exists because they are not tied to a specific instance. They are shared across all instances of the class.

3. Can we apply static keyword with a top-level class?

Ans: No, static keyword cannot be applied with outer or top-level class but an inner class can be static.

4. Will the following code snippet compile successfully? If yes, what is the output of the following program?

```
public class Myclass
{
    private int x = 10;

    static int m1() {
        int y = x;

        return y;
    }
}
```

```
}  
  
public static void main(String[] args) {  
  
    m1();  
  
}  
  
}
```

Ans: No, the above code will not be compiled because x is an instance variable and instance member cannot be accessed from static region.

5. Identify the error in the following code snippet. If there is no error then what will be the output of the program?

```
public class Myclass  
{  
  
    private int x = 10;  
  
    static int m1()  
  
    {  
  
        Myclass obj = new Myclass();  
  
        int y = obj.x;  
  
        return y;  
  
    }  
  
    public static void main(String[] args) {  
  
        System.out.println(m1());  
  
    }  
  
}
```

Ans: There is no error in the above code snippet. Output: 10.

6. What is the main use of static keyword in java?

Ans: The main use of java static keyword is as follows:

- The static keyword can be used when we want to access the data, method, or block of the class without any object creation.
- It can be used to make the programs more memory efficient.

7. Can we mark a local variable as static?

Ans: No, we cannot mark a local variable with a static keyword.

8. When does a static variable get memory?

Ans: When a class is loaded into the memory at runtime, the static variable is created and initialized into the common memory location only once.

9. What will be the output of the following program?

```
public class Myclass
```

```
{
```

```
    static int a = 20;
```

```
    static int b = 30;
```

```
    static int c = 40;
```

```
    Myclass()
```

```
{
```

```
    a = 200;
```

```
}
```

```
static void m1() {
```

```
    b = 300;
```

```
}
```

```
static {
```

```
    c = 400;
```

```
}  
  
public static void main(String[] args) {  
  
    System.out.println(a);  
  
    System.out.println(b);  
  
    System.out.println(c);  
  
}  
  
}
```

Ans: Output: 20, 30, 400.

10. What will be the output of the following code?

```
public class Myclass {  
  
    static int a = 20;  
  
    Myclass() {  
  
        a = 200;  
  
    }  
  
    public static void main(String[] args) {  
  
        new Myclass();  
  
        System.out.println(a);  
  
    }  
  
}
```

Ans: Output: 200.

11. In which part of memory static variables are stored?

Ans: All static variables are stored in PermGen space of the heap memory.

12. How static variable is different from the instance variable?

Ans: The difference between static variable and instance variable is as follows:

- a) A static variable is also called class variable whereas, an instance variable is also called non-static variable.
- b) Class variable can be accessed inside a static block, instance block, static method, instance method, and method of the inner class whereas, instance variable can be accessed only inside the instance members, and method of the inner class.
- c) Class variable is always resolved during compile time whereas, instance variable is resolved during the runtime.
- 4) Static variable cannot be serialized in Java whereas, instance variable can be serialized.

13. Will the following code snippet compile fine? If yes, what will be the output of the following program?

```
public class Myclass {  
  
    static int a = 20;  
  
    Myclass() {  
  
        a++;  
  
    }  
  
    void m1() {  
  
        a++;  
  
        System.out.println(a);  
  
    }  
  
    public static void main(String[] args)
```

```
{  
  
    Myclass obj = new Myclass();  
  
    Myclass obj2 = new Myclass();  
  
    Myclass obj3 = new Myclass();  
  
        obj3.m1();  
  
}  
  
}
```

Ans: Output: 24.

14. What is a static method in Java?

Ans: When a method is declared with the keyword 'static', it is called static method in java.

15. Why is a static method also called a class method?

Ans: A static method is also called a class method because it ties to a class rather than an individual instance of a class. Therefore, we need not to create an object of the class to call and execute static method.

16. Can we access static members (such as static variables and static methods) from an instance method?

Ans: Yes, we can access static members from an instance method in java.

17. Is it possible to access instance members from a static method?

Ans: No, it is not possible to access instance members like instance variable and instance method from a static method.

18. Identify the error in the following code.

```
public class Test  
  
{  
  
    Test() {  
  
        m2();  
    }  
}
```

```
}

void m1() {

    System.out.println("Instance method");

}

static void m2() {

    System.out.println("Static method");

    m1();

}

public static void main(String[] args)

{

    new Test();

}

}
```

Ans: There is an error inside the static method because we cannot make a static reference to the non-static method m1() from the type Test.

19. Will the below code compile successfully? If yes, what will be the output of the following program?

```
public class Test

{

    Test(Test t) {

        m1();

        System.out.println("Constructor");

    }

}
```

```
void m1() {  
  
    m2();  
  
    System.out.println("Instance method");  
  
}  
  
static void m2() {  
  
    System.out.println("Static method");  
  
}  
  
public static void main(String[] args)  
{  
  
    new Test(null);  
  
}  
}
```

Ans: Yes, the above code will be compiled successfully. There is no problem.
Output: Static method, Instance method, Constructor.

20. Is there any error in the below code snippet? If yes, identify the error and give the reason behind it.

```
public class Test  
{  
  
    void m1(Test test) {  
  
        System.out.println("Instance method");  
  
    }  
  
    static void m1(Test t) {  
  
        System.out.println("Static method");  
  
    }  
}
```



```
}  
  
}
```

Ans: Yes, Duplicate method error. This is because we cannot declare a static method and instance method with the same signature in the same class.

21. What is the difference between static method and instance method?

Ans: Go to this tutorial: [Static method in Java](#)

22. Can we have a static method in an interface?

Ans: Yes, from Java 8 and onwards, the interface allows to define a static method with body.

23. Can we use this or super keyword in static method in Java?

Ans: No, In the entire core java, this and super keywords are not allowed inside the static region.

24. Is it possible to overload static methods in a class?

Ans: Yes, we can overload static methods but override them. This is because they are bound with class, not instance.

25. Is it possible to override static methods of a class?

Ans: No, we cannot override static methods because static methods belong to a class, not individual objects, and are resolved at compile time by java compiler.

For more detail, go to this tutorial: [Can we override static methods in Java?](#)

26. Can we override an instance method as static?

Ans: No.

27. What is the output of the following program code?

```
public class Myclass {  
  
    static int a = 20;  
  
    static void m2() {
```

```
        a++;  
  
    }  
  
    public static void main(String[] args) {  
  
        System.out.println(a);  
  
    }  
  
}
```

Ans: Output: 20.

28. Why static block is executed before the main method in java?

Ans: When the dot class file is loaded into memory, static block is executed. After executing the static block, JVM calls the main method to start execution. Therefore, static block is executed before the main method.

29. What is the output of the following program below?

```
public class Myclass {  
  
    static int a = 20;  
  
    static {  
  
        a++;  
  
    }  
  
    {  
  
        a++;  
  
        System.out.println(a);  
  
    }  
  
    public static void main(String[] args)
```

```
{  
  
    Myclass obj = new Myclass();  
  
    Myclass obj2 = new Myclass();  
  
    Myclass obj3 = new Myclass();  
  
}  
  
}
```

Ans: Output: 22, 23, 24.

30. What is the use of static block in java?

Ans: A static block can be used when

- we want to write that logic inside static block that is executed during the class loading.
- we want to change the default value of static variables.
- we want to initialize static variable of the class.

31. What is the output of the following code snippet?

```
public class Myclass {  
  
    Myclass() {  
  
        System.out.println("constructor");  
  
    }  
  
    static void m1() {  
  
        System.out.println("static method");  
  
    }  
  
    void m2(){  
  
        System.out.println("instance method");  
  
    }  
  
}
```

```
}  
  
static {  
  
    System.out.println("static block");  
  
}  
  
{  
  
    System.out.println("instance block");  
  
}  
  
public static void main(String[] args)  
{  
  
    Myclass obj = new Myclass();  
  
    m1();  
  
    obj.m2();  
  
}  
  
}
```

Ans: Output: static block, instance block, constructor, static method, instance method.

32. How static block is different from an instance block in java?

Ans: Static block is different from an instance block by the following key points:

- a) Static block is also called a static initialization block whereas instance block is also called instance initialization block or non-static block.
- b) Static block gets executed before the instance block whereas, instance block executes after the static block.

- c) Only static variables can be accessed inside the static block whereas, both static and non-static variables can be accessed inside the instance block.
 - d) Static block executes when the class is loaded into the memory whereas instance block executes only when an instance of the class is created.
 - e) We cannot use this keyword inside the static block whereas this keyword can be used in the instance block.
-

33. Can we declare a static block inside a method?

Ans: No, we cannot declare a static block inside a method.

34. What will be the output of the following code snippet after execution?

```
public class Myclass {  
  
    static {  
  
        System.out.println("static block");  
  
    }  
  
    {  
  
        System.out.println("instance block");  
  
    }  
  
    public static void main(String[] args) {  
  
        Myclass obj = new Myclass();  
  
        Myclass obj2 = new Myclass();  
  
        Myclass obj3 = new Myclass();  
  
    }  
  
}
```

Ans: Output: static block, instance block, instance block, instance block.

35. Will the following code snippet compile fine? If yes, what will be output after execution?

```
public class Myclass {  
  
    private static int x = 10;  
  
    static {  
  
        x++;  
  
    }  
  
    static {  
  
        ++x;  
  
    }  
  
    {  
  
        x--;  
  
    }  
  
    public static void main(String[] args) {  
  
        Myclass obj = new Myclass();  
  
        Myclass obj2 = new Myclass();  
  
        Myclass obj3 = new Myclass();  
  
        System.out.println(x);  
  
    }  
  
}
```

Ans: Yes, the code will be compiled fine. The output is 9.

Hope that this tutorial has covered almost all the important **static keyword interview questions in java** with nice answers. I hope that you will have understood answers to all java static interview questions and prepared them.

All the best!!!

