

Java Interview Question and Answers

➤ Topic – Java String:

Java String Handling Interview Questions with Answers

1. What is String in Java?

Ans: In Java, String is an object that represents a sequence of characters. For example, "Pencil" is a string of 6 characters. String is not a primitive data type like int and long.

2. Is String a class or data type in Java?

Ans: String is a class that is present in java.lang package. But we also considered all classes in java as data types. So, we can say a string as data type also.

3. How many ways can we create string object in Java?

Ans: Generally, there are two ways to create the string object in java. They are:

1. By string literal
2. By new keyword

String literal: String literal is created by using double quotes. For example:
`String s = "Hello";`

The string literal always creates in the string constant pool. Whenever we create a string object using string literal, JVM checks string constant pool first.

If string already exists in the string constant pool, JVM will not create a new string object in string constant pool. JVM will point to the same string object by a reference to save memory.

But if the string does not exist in the string pool, JVM creates a new string object and store it in the pool.

New keyword: It is the second way of creating string object in java. It is just like creating an object of any class. We can declare it as:

```
String s = new String("Hello");
```

When we use new keyword to create an object of string class, JVM will create two objects. First, it will create an object in Heap memory and stores string "Hello" into the object and then will point a reference to object in the heap.

Now, JVM will create the second object as a copy for literal "Hello" in string constant pool for future purposes. There is no explicit reference variable pointing to the copy object in the pool.

4. What is String Constant Pool in Java?

Ans: String constant pool is a special memory area in heap which is used for storing string objects. Internally, the string class uses a string constant pool.

5. What is the meaning of Immutable in terms of String?

Ans: The meaning of immutable is unchangeable or unmodifiable. It means that once string object with value has created, its value cannot change or modify.

But if we try to change with a new value, a new string object will be created by storing a new value. Hence, we cannot perform any changes with existing string object. This concept is called immutability in java.

6. Why string object is immutable in java?

Ans: String class is immutable and final in java because of several benefits. They are:

a) String is immutable in java because java uses the concept of string literal. Suppose there are five reference variables, all refer to the same object "Hello".

If one reference variable of them changes the value of an object, it will affect all reference variables.

b) Java implements this immutability concept to minimize the duplication of string value.

c) It provides security because any hacker can not change its value. It helps to store sensitive information such as database username and password.

d) It is safe to use in multi-threading. So, we do not need any synchronization.

7. What is the superclass of string class in Java?

Ans: Object class is the superclass of string. String class extends object class.

8. What is String literal in java? How is string literal stored in memory?

Ans: String literal is a concept of Java language where string class is optimized to cache all the strings created in double-quotes. They are stored in string constant pool.

9. Why Java uses the concept of string literal?

Ans: Java uses string literal concept to make it more memory efficient because no new object will create if it already exists in the constant pool.

10. How are String literals cleared from String constant pool?

Ans: It is the responsibility of garbage collector to clear string object from the string constant pool.

11. What are the drawbacks of creating a large number of string literals in the memory pool?

Ans: Creating a large number of string literals in the memory pool can lead to memory leaks and performance issues.

12. How many objects will create for the identical string?

Ans: Two identical string literal would create two separate string references but both will refer to the same object because string class is immutable in java.

13. How many total objects will be created in the following code?

```
String s1 = "Hello";  
  
String s2 = "Hello";  
  
String s3 = "Hello";
```

Ans: Only one object will be created in the string constant pool.

14. How many total objects will create in the following code?

```
String s = new String("Hello");
```

Ans: A total of two objects will create, one in the heap area and another in string constant pool.

15. How many total objects will be created in the following code?

```
String s1 = new String("Sciencetech");  
  
String s2 = new String("Sciencetech");  
  
String s3 = "Sciencetech";  
  
String s4 = "Sciencetech";
```

Ans: A total of three objects will be created, two in the heap area and one in string constant pool.

16. Can we use reflection to clear a string object?

Ans: Yes, we can use reflection to clear string object from the memory but it is not recommended to do.

17. Why string class is declared as final in java?

Ans: String class has been marked as final so that we could not override the immutable behavior of string class.

18. How many interfaces are implemented by String class?

Ans: String class implements three interfaces as Serializable, Comparable, and CharSequence.

19. Does String is thread-safe in java?

Ans: Yes, String is thread-safe in java because string is immutable. So, we cannot change its value once created in the program.

This means it is thread-safe and can be safely used in a multi-threading environment. No synchronization is required for the string objects.

20. What is the disadvantage of string class in java?

Ans: we cannot extend String class to get additional features.

21. Is String a wrapper class?

Ans: No, String is not a wrapper class.

22. How to compare two Strings in Java?

Ans: In Java, two strings can be compared on the basis of content and reference. Basically, there are three general ways by which we can compare strings.

- Using equals() method
- By == operator (double equal operators)
- By compareTo() method.

23. What is the difference between == (double equal operator) and equals method in Java?

Ans: The difference between double equal operator and equals method are as follows:

a) The == (double equal operator) compares two object references, not characters. i.e. it compares the memory address of the object references.

Whereas, string equals() method is used to compare the original content of the string. It compares the two strings for equality.

b) In double equal operator, if both references are pointing to the same object then it will return true whereas, in equals method, if all the characters are the same, it will return true.

c) If both references are pointing to the different objects, it will automatically return false, whereas, if any character is not matching, it returns false.

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

```
public class Test {  
  
    public static void main(String args[])  
  
    {  
  
        String s1 = "Hello";  
  
        String s2 = "Hello";  
  
        String s3 = new String("Good bye");  
  
        String s4 = new String("Hello");  
    }  
}
```

```
System.out.println(s1.equals(s2));

System.out.println(s1.equals(s3));


System.out.println(s1.equals(s4));

System.out.println(s1.equals(args));

System.out.println(s1.equals(null));

}

}
```

Ans: The output of following code is as follows:

- True because content and case are the same.
- False because content is not the same.
- True because content and case are the same.
- False
- False

25. What will be the output of the below code?

```
public class Test {

public static void main(String args[])

{

String s1 = "GOOD BYE";

String s2 = new String("Good bye");


System.out.println(s1.equals(s2));

System.out.println(s1.equalsIgnoreCase(s2));

}
```

```
}
```

Ans: Output:

- False because content is the same but the case is different.
- True

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

```
public class Test {  
  
    public static void main(String args[])  
  
    {  
  
        String s1 = "Cricket";  
  
        String s2 = "Cricket";  
  
        String s3 = new String("Cricket");  
  
  
        System.out.println(s1==s2);  
  
        System.out.println(s1==s3);  
  
    }  
  
}
```

Ans: The output of this program is as follows:

- True because s1 and s2 both are pointing to the same object created in the SCP area.
- False because both have a different reference number.

27. Consider the following code and think about the output.

```
public class StringCompareTest {  
  
    public static void main(String[] args)  
  
    {  
  
        String st1 = "Ivaan";
```

```
String st2 = "Hilery";

String st3 = "Ivaan";

String st4 = new String("Ivaan");

System.out.println(st1.compareTo(st2));

System.out.println(st1.compareTo(st3));

System.out.println(st3.compareTo(st1));

System.out.println(st2.compareTo(st4));

}

}
```

Output:

```
1
0
0
-1
```

28. What are the best ways to concatenate strings in Java?

Ans: There are two convenient and *best ways to concatenate strings in Java*.

They are:

- By concat() method: Concatenates or joins the specified string to the end of current string and creates a new string object.
- By + (String concatenation) operator: Used to add two or more strings.

29. What will be the output of the following Java program?

a)

```
public class StringConTest {

public static void main(String[] args)
```



```
{  
  
String s1 = "Java";  
  
s1.concat("Programming");  
  
System.out.println(s1);  
  
}  
  
}
```

Output:

Java

b)

```
public class StringConTest {  
  
public static void main(String[] args)  
  
{  
  
String s1 = new String("Java");  
  
s1.concat(" Core");  
  
s1 = s1.concat(" Technology");  
  
System.out.println(s1);  
  
}  
  
}
```

Output:

Java Technology

c)

```
public class StringConTest {
```

```
public static void main(String[] args)
{
    String s1 = 25 + 25 + " Text " + 1 + 4;

    String s2 = new String("Text2");

    System.out.println(s1);

    System.out.println(s1.concat(s2));

}
}
```

Output:

50 Text 14

50 Text 14Text2

30. Consider the following program code.

```
public class SubStringTest {

    public static void main(String[] args)
    {

        String s = new String("Java Technology");

        s.substring(5);

        System.out.println(s);

        String s2 = s.substring(6, 15);

        System.out.println(s2);

    }

}
```

a) What will be the output of this program?

b) How many string objects will create in the heap and string constant pool?

Ans a: The output of the above program is:

```
Java Technology
```

```
echnology
```

Ans b: A total of 4 objects will create, one in the string constant pool and three in the heap area.

31. How to check a string is empty or not?

Ans: String class provides isEmpty() method to check the string is empty or not. If the length of the string is 0, the string is empty and will return true otherwise false. For example:

```
public class StringEmptyTest {  
  
    public static void main(String[] args)  
  
    {  
  
        String str = "";  
  
        boolean isEmpty1 = str.isEmpty();  
  
        System.out.println("Is String empty: " +isEmpty1);  
  
  
        String str2 = " ";  
  
        boolean isEmpty2 = str2.isEmpty();  
  
        System.out.println("Is String empty: " +isEmpty2);  
  
    }  
  
}
```

Output:

Is String empty: true

Is String empty: false

32. What will be the output of Java program?

```
public class StringUpperCaseTest {  
  
    public static void main(String[] args)  
  
    {  
  
        String s = new String("text");  
  
        String s2 = "text";  
  
        String s3 = s.toUpperCase();  
  
        System.out.println(s==s3);  
  
        System.out.println(s.equals(s2));  
  
        System.out.println(s2.equalsIgnoreCase(s3));  
  
    }  
  
}
```

Output:

false

true

true

33. How to get a character or multiple characters from a string in Java?

Ans: Refer to this tutorial: [Get Character from String in Java](#)

34. What are the most widely used methods of Java String class?

Ans: There are following most widely used methods of String class. They are:

- concat()

- length()
- replace()
- substring()
- toString()
- trim()
- toUpperCase()
- toLowerCase()
- equals()
- equalsIgnoreCase()
- charAt()

To know more about these methods of String class, go to this tutorial: [Java String Class](#)

StringBuffer Interview Questions in Java

35. What is a mutable string in Java?

Ans: A string that can be modified or changed into the existing object is known as mutable string. StringBuffer and StringBuilder classes are used to create a mutable string.

36. Why StringBuffer objects in Java are mutable?

Ans: StringBuffer objects in Java are mutable because they allow to create strings of flexible length that can be modified in terms of both length and content. It means that once we create a StringBuffer class object, we can perform any required changes in the object. i.e., we can change its data.

37. What is the difference between length and capacity in Java StringBuffer?

Ans: Capacity is the total number of characters hold in the StringBuffer object. Whereas, length is the number of characters already present in the StringBuffer object.

38. How will you add string in StringBuffer? Give an example.

Ans: StringBuffer class provides append() method to add string in StringBuffer. The code is as:

```
StringBuffer sb = new StringBuffer();  
  
sb.append("Java");  
  
sb.append("Programming");
```

39. When will you use StringBuffer if String class is already available?

Ans: There are some such cases where we can not recommend using a string concept. For example:

- a) When you have fixed content and will not change frequently, then use String class.
- b) When you have no fixed content, and frequently changing, then use the StringBuffer class.

40. What are the most widely used methods of StringBuffer class in Java?

Ans: The most widely used methods of StringBuffer class is as:

- append()
- capacity()
- charAt()
- delete()
- ensureCapacity()
- getChars()
- indexOf()
- insert()
- length()
- reverse()

To get more detail about these methods with example programs, go to this tutorial: [StringBuffer Methods in Java](#)

StringBuilder Interview Questions in Java

41. Why do we need StringBuilder in Java?

Ans: In Java, StringBuilder class is identical to StringBuffer class except for one important difference:

StringBuffer is thread-safe and StringBuilder is not thread-safe. Methods provided by StringBuffer for modifying the buffer are synchronized, so only one thread may access StringBuffer object at a time.

Most of the time, we do not require thread safety. In such a case, if we use StringBuffer, performing the application will be down. That's why J2SE 5 added StringBuilder class later.

42. Which class is more preferable: StringBuffer or StringBuilder? Why?

Ans: Methods of StringBuilder are not synchronized. They can execute faster than StringBuffer. Therefore, StringBuilder is more preferable than StringBuffer when you are not constructing thread-safe application.

43. When should you use StringBuilder?

Ans: You can use StringBuilder when you want to create a string within a single thread. It will improve the execution time and performance. StringBuilder is suitable to use when you do not need thread-safety.

44. What is the difference between String, StringBuffer, and StringBuilder?

Or. **what is the difference between StringBuffer and StringBuilder?**

Ans: For best answer, go to this tutorial: [String vs StringBuffer vs StringBuilder](#)

45. Look at the below code and explain how many reference variables and how many string objects will create in the memory?

```
String str1 = "Technology"; // Line 1  
  
String str2 = str1;  
  
str1 = str1.concat("Java"); // Line 2
```

Ans: This code will create two reference variables and two string objects in the memory. The two reference variables are str1 and str2. The two string objects are: Technology, and TechnologyJava.

46. What will be the output when you compile and execute the below code?

```
public class Test {
```

```
public static void main(String[] args)
{
    String str1 = "xyz";

    String str2 = str1.concat("pqr").toUpperCase().replace("Y", "d"); // Line 1

    System.out.println(str2);
}
}
```

Ans: On compiling and executing the above code, the value of str2 will be XdZPQR. It is an example of chained methods. In line 1, the expression will execute from left to right.

First, str1 is concatenated to pqr and so the value is xyzpqr. Then, the value is converted into uppercase and the value becomes XYZPQR. At last, the character 'Y' is replaced with d and the value becomes XdZPQR.

47. What is immutable class in Java?

Ans: An immutable class in Java is a class whose state of an object cannot be changed or modified after it is created. All the wrapper classes in java like Integer, Long, Short, Byte, Boolean, Float, and Double are immutable class.

48. What is a mutable class and mutable object in Java?

Ans: A mutable class in Java is a class whose objects are mutable (i.e., modifiable). If the state of an object can be changed or mutated after it has been created, it is called mutable object in java.

Java.util.Date, StringBuffer and StringBuilder are the examples of mutable classes in Java.

49. How to create/make an Immutable Class in Java?

Ans: To get an exact answer, go to this tutorial: [Immutable class in Java](#)

50. What is the use of toString() method in Java?

Ans: The toString() method of String class returns the string representation of any object. Java compiler internally calls toString() method on the object if you print any object.

51. What is the purpose of intern() method in Java?

Ans: The purpose of intern() method is to add the unique copy of the string object to the string constant pool manually. When we create a string using a new keyword, JVM stores it in the heap memory and also store the unique copy of that string object in the string pool using the intern() method.

When we do the same thing again, JVM will check if the string object with the same content is present in the string constant pool or not.

If a string object with the same content is present, JVM will simply point the reference of that object to the respective string variable.

If a string object with the same content is not present in the string pool, JVM creates a string object with the same content in the String pool and returns its reference to the string variable.

52. What is the purpose of format() method in Java String?

Or, what is the difference between format() method and printf() method?

Ans: The main purpose of both format() and printf() methods is to format the string. The only difference is that the format() method returns the formatted string, whereas printf() method prints the formatted string.

That's why, if you want the formatted string in the program, then use the format() method. And if you want to just print the formatted string, use the printf() method.

53. Is String is 'thread-safe' in Java?

Ans: Yes, the string is thread-safe in Java.

54. Is it possible to convert String to Int and vice versa in Java?

Ans: Yes, it is possible to convert string to int and vice versa. We can convert string to an integer using parseInt() and valueOf() methods of the Integer class.

Also, we can convert an integer to string using valueOf() method of Java String class. Look at the source code that demonstrates the string to integer and integer to string conversion.

```
public class Conversion {
```

```
public static void main(String[] args)

{

String str = "1456";

int num = 9878;


// Converting string to int using Integer.parseInt() method

int n1 = Integer.parseInt(str);

// Converting string to int using Integer.valueOf() method

int n2 = Integer.valueOf(str);


System.out.println("Converting String into Integer:");

System.out.println("Using Integer.parseInt() method: " +n1);

System.out.println("Using the Integer.valueOf() method : " +n2);


System.out.println();

// Converting integer to string using String.valueOf() method

String s = String.valueOf(num);

System.out.println("Converting Integer to String:");

System.out.println("Using the String.valueOf() method : "+s);

}

}
```

Output:

Converting String into Integer:

Using Integer.parseInt() method: 1456

Using the Integer.valueOf() method : 1456

Converting Integer to String:

Using the String.valueOf() method : 9878

55. What is the split() method? Can we spilt a string in Java?

Ans: String class provides a split() method that split the string based on the provided a regular expression delimiter. This method returns an array of the split substrings. The signature of split() method is:

```
public String[] split(String regex)
```

The example code demonstrates the use of split method.

```
public class SplitExample {  
  
    public static void main(String[] args)  
    {  
  
        String name = "What, is, your, name?";  
  
        String[ ] substringArray = name.split(","); // Splitting string based on comma.  
  
        for(String substring : substringArray)  
        {  
  
            System.out.println(substring);  
  
        }  
  
    }  
}
```

```
}
```

Output:

What

is

your

name?

56. What is StringJoiner in Java 8?

Ans: StringJoiner is a new class added in Java 8. It joins (or concatenates) Strings separated by a delimiter and having a prefix and suffix. The following code shows this:

```
import java.util.StringJoiner;

public class StringJoiner {

    public static void main(String[] args)

    {

        StringJoiner strJoiner = new StringJoiner(",", "[", "]");

        strJoiner.add("Red");

        strJoiner.add("Green");

        strJoiner.add("Blue");

        System.out.println(strJoiner);

    }

}
```

The above code creates a StringJoiner. It uses comma symbol (,) as delimiter and square brackets ([]) as prefix and suffix. Then, it calls add()

method with some string values. Hence, this code displays the following output:

Output:

```
[Red,Green,Blue]
```

57. Is it possible to get a Character Array from String in Java?

Ans: Yes, it is possible to get a character array from string in Java. String class provides `toCharArray()` method, which converts this string to a new character array. This method returns a copy of its internal char array, whose length is the length of this string.

Hope that this tutorial has covered all the important **Java string interview questions** with the best possible answers. These string handling interview questions in Java are very important for technical tests and interviews. You must prepare the answers of these string interview questions before going to interviews. Our team will update regularly with more important interview questions.

All the best!!!

