Java String Interview Questions

What is String in Java? String is a data type?

String is a Class in java and defined in java.lang package. It’s not a primitive data type like int and long. String class represents character Strings. String is used in almost all the Java applications and there are some interesting facts we should know about String. String in immutable and final in Java and JVM uses String Pool to store all the String objects. Some other interesting things about String is the way we can instantiate a String object using double quotes and overloading of “+” operator for concatenation.

What are different ways to create String Object?

We can create String object using new operator like any normal java class or we can use double quotes to create a String object. There are several constructors available in String class to get String from char array, byte array, StringBuffer and StringBuilder.

```
String str = new String("abc");
String str1 = "abc";
```

When we create a String using double quotes, JVM looks in the String pool to find if any other String is stored with same value. If found, it just returns the reference to that String object else it creates a new String object with given value and stores it in the String pool.
When we use new operator, JVM creates the String object but don’t store it into the String Pool. We can use intern() method to store the String object into String pool or return the reference if there is already a String with equal value present in the pool.

Write a method that will remove given character from the String?

We can use replaceAll method to replace all the occurance of a String with another String. The important point to note is that it accepts String as argument, so we will use Character class to create String and use it to replace all the characters with empty String.

```
private static String removeChar(String str, char c) {
    if (str == null)
        return null;
    return str.replaceAll(Character.toString(c), "");
}
```

How can we make String upper case or lower case?

We can use String class toUpperCase and toLowerCase methods to get the String in all upper case or lower case. These methods have a variant that accepts Locale argument and use that locale rules to convert String to upper or lower case.

Write a method to check if input String is Palindrome?

A String is said to be Palindrome if it’s value is same when reversed. For example “aba” is a Palindrome String.
String class doesn’t provide any method to reverse the String but StringBuffer and StringBuilder class has reverse method that we can use to check if String is palindrome or not.

```
private static boolean isPalindrome(String str) {
    if (str == null)
        return false;
    StringBuilder strBuilder = new StringBuilder(str);
    strBuilder.reverse();
    return strBuilder.toString().equals(str);
}
```

Sometimes interviewer asks not to use any other class to check this, in that case we can compare characters in the String from both ends to find out if it’s palindrome or not.

```
private static boolean isPalindromeString(String str) {
    if (str == null)
        return false;
    int length = str.length();
    System.out.println(length / 2);
    for (int i = 0; i < length / 2; i++) {

        if (str.charAt(i) != str.charAt(length - i - 1))
            return false;
    }
    return true;  }
```

What is String subSequence method?

Java 1.4 introduced CharSequence interface and String implements this interface, this is the only reason for the implementation of subSequence method in String class. Internally it invokes the String substring method.
Check this post for String subSequence example.

How to compare two Strings in java program?

Java String implements Comparable interface and it has two variants of compareTo() methods.
compareTo(String anotherString) method compares the String object with the String argument passed lexicographically. If String object precedes the argument passed, it returns negative integer and if String object follows the argument String passed, it returns positive integer. It returns zero when both the String have same value, in this case equals(String str) method will also return true.
compareToIgnoreCase(String str): This method is similar to the first one, except that it ignores the case. It uses String CASE_INSENSITIVE_ORDER Comparator for case insensitive comparison. If the value is zero then equalsIgnoreCase(String str) will also return true.
Check this post for String compareTo example.

How to convert String to char and vice versa?

This is a tricky question because String is a sequence of characters, so we can’t convert it to a single character. We can use use charAt method to get the character at given index or we can use toCharArray() method to convert String to character array.
Check this post for sample program on converting String to character array to String.

How to convert String to byte array and vice versa?

We can use String getBytes() method to convert String to byte array and we can use String constructor new String(byte[] arr) to convert byte array to String. Check this post for String to byte array example.

Can we use String in switch case?

This is a tricky question used to check your knowledge of current Java developments. Java 7 extended the capability of switch case to use Strings also, earlier java versions doesn’t support this. If you are implementing conditional flow for Strings, you can use if-else conditions and you can use switch case if you are using Java 7 or higher versions. Check this post for Java Switch Case String example.

Write a program to print all permutations of String?

This is a tricky question and we need to use recursion to find all the permutations of a String, for example “AAB” permutations will be “AAB”, “ABA” and “BAA”.
We also need to use Set to make sure there are no duplicate values.
Check this post for complete program to find all permutations of String.

Write a function to find out longest palindrome in a given string?

A String can contain palindrome strings in it and to find longest palindrome in given String is a programming question.
Check this post for complete program to find longest palindrome in a String.

Difference between String, StringBuffer and StringBuilder?

String is immutable and final in java, so whenever we do String manipulation, it creates a new String. String manipulations are resource consuming, so java provides two utility classes for String manipulations – StringBuffer and StringBuilder.
StringBuffer and StringBuilder are mutable classes. StringBuffer operations are thread-safe and synchronized where StringBuilder operations are not thread-safe. So when multiple threads are working on same String, we should use StringBuffer but in single threaded environment we should use StringBuilder.
StringBuilder performance is fast than StringBuffer because of no overhead of synchronization.

Check this post for extensive details about String vs StringBuffer vs StringBuilder.
Read this post for benchmarking of StringBuffer vs StringBuilder.

Why String is immutable or final in Java?

There are several benefits of String because it’s immutable and final.
String Pool is possible because String is immutable in java.
It increases security because any hacker can’t change its value and it’s used for storing sensitive information such as database username, password etc.
Since String is immutable, it’s safe to use in multi-threading and we don’t need any synchronization.
Strings are used in java classloader and immutability provides security that correct class is getting loaded by Classloader. Check this post to get more details why String is immutable in java.

How to Split String in java?

We can use split(String regex) to split the String into String array based on the provided regular expression.
Learn more at java String split.

Why Char array is preferred over String for storing password?

String is immutable in java and stored in String pool. Once it’s created it stays in the pool until unless garbage collected, so even though we are done with password it’s available in memory for longer duration and there is no way to avoid it. It’s a security risk because anyone having access to memory dump can find the password as clear text.
If we use char array to store password, we can set it to blank once we are done with it. So we can control for how long it’s available in memory that avoids the security threat with String.

How do you check if two Strings are equal in Java?

There are two ways to check if two Strings are equal or not – using “==” operator or using equals method. When we use “==” operator, it checks for value of String as well as reference but in our programming, most of the time we are checking equality of String for value only. So we should use equals method to check if two Strings are equal or not.
There is another function equalsIgnoreCase that we can use to ignore case.

```
String s1 = "abc";
String s2 = "abc";
String s3= new String("abc");
System.out.println("s1 == s2 ? "+(s1==s2)); //true
System.out.println("s1 == s3 ? "+(s1==s3)); //false
System.out.println("s1 equals s3 ? "+(s1.equals(s3))); //true
```

What is String Pool?

As the name suggests, String Pool is a pool of Strings stored in Java heap memory. We know that String is special class in java and we can create String object using new operator as well as providing values in double quotes. Check this post for more details about String Pool.

What does String intern() method do?

When the intern method is invoked, if the pool already contains a string equal to this String object as determined by the equals(Object) method, then the string from the pool is returned. Otherwise, this String object is added to the pool and a reference to this String object is returned.
This method always return a String that has the same contents as this string, but is guaranteed to be from a pool of unique strings.

Does String is thread-safe in Java?

Strings are immutable, so we can’t change it’s value in program. Hence it’s thread-safe and can be safely used in multi-threaded environment.
Check this post for Thread Safety in Java.

Why String is popular HashMap key in Java?

Since String is immutable, its hashcode is cached at the time of creation and it doesn’t need to be calculated again. This makes it a great candidate for key in a Map and it’s processing is fast than other HashMap key objects. This is why String is mostly used Object as HashMap keys.

String Programming Questions :

What is the output of below program?

```
package com.journaldev.strings;

    public class StringTest {

        public static void main(String[] args) {
            String s1 = new String("pankaj");
            String s2 = new String("PANKAJ");
            System.out.println(s1 = s2);
        }

    }
```

It’s a simple yet tricky program, it will print “PANKAJ” because we are assigning s2 String to s1. Don’t get confused with == comparison operator.

What is the output of below program?

```
package com.journaldev.strings;

public class Test {

    public void foo(String s) {
    System.out.println("String");
    }

    public void foo(StringBuffer sb){
    System.out.println("StringBuffer");
    }

    public static void main(String[] args) {
      new Test().foo(null);
   }
   }
```

The above program will not compile with error as “The method foo(String) is ambiguous for the type Test”. For complete clarification read Understanding the method X is ambiguous for the type Y error.

What is the output of below code snippet?

```
String s1 = new String("abc");
String s2 = new String("abc");
System.out.println(s1 == s2);
```

It will print false because we are using new operator to create String, so it will be created in the heap memory and both s1, s2 will have different reference. If we create them using double quotes, then they will be part of string pool and it will print true.

What will be output of below code snippet?

```
String s1 = "abc";
StringBuffer s2 = new StringBuffer(s1);
System.out.println(s1.equals(s2));
```

It will print false because s2 is not of type String. If you will look at the equals method implementation in the String class, you will find a check using instanceof operator to check if the type of passed object is String? If not, then return false.

Does String is thread-safe in Java?

If you are familiar with the concept of immutability and thread-safety you can easily answer this String interview question in Java. Since String is immutable, it is thread-safe and it can be shared between multiple thread without external synchronization.

What will be output of below program?

```
String s1 = "abc";
String s2 = new String("abc");
s2.intern();
System.out.println(s1 ==s2);
```

It’s a tricky question and output will be false. We know that intern() method will return the String object reference from the string pool, but since we didn’t assigned it back to s2, there is no change in s2 and hence both s1 and s2 are having different reference. If we change the code in line 3 to s2 = s2.intern(); then output will be true.

How many String objects got created in below code snippet?

```
String s1 = new String("Hello");
String s2 = new String("Hello");
```

Answer is 3.
First – line 1, “Hello” object in the string pool.
Second – line 1, new String with value “Hello” in the heap memory.
Third – line 2, new String with value “Hello” in the heap memory. Here “Hello” string from string pool is reused.

What is String in Java? Is String is data type?
String in Java is not a primitive data type like int, long or double. String is a class or in more simple term a user defined type. This is confusing for some one who comes from C background. String is defined in java.lang package and wrappers its content in a character array. String provides equals() method to compare two String and provides various other method to operate on String like toUpperCase() to convert String into upper case, replace() to replace String contents, substring() to get substring, split() to split long String into multiple String.

Why String is final in Java?
String is final by design in Java, some of the points which makes sense why String is final is Security, optimization and to maintain pool of String in Java. for details on each of this point see Why String is final in Java.

What is Difference between String and StringBuffer in Java??
This is probably the most common question on String I have seen in Java interviews. Though String and StringBuffer are two different class they are used in context of concatenating two Strings, Since String is immutable in Java every operation which changes String produces new String, which can be avoided by using Stringbuffer. See String vs StringBuffer for more details.

What is difference in String on C and Java?
If you have mentioned C in your resume, then you are likely to face this String interview question. Well C String and Java String are completely different to each other, C String is a null terminated character array while String in Java is an Object. Also String is more feature rich in Java than C.

Why char array is better than String for storing password?
This String interview question is debatable and you might not agree with interviewer but this is also a chance to show that how deep and differently you can think of. One of the reason which people give Why you should store password in char array over String is related to immutability, since its not possible to remove erase contents of String but you can erase contents of char array. See Why char array preferred over String for password for complete discussion.

How do you compare two String in Java?
This is another common String interview question which appears on fresher level interviews. There are multiple ways to compare two String like equals() method, equalsIgnoreCase() etc, You can also see 4 ways to compare String in Java for more examples. Main thing which interviewer checks is that whether candidate mentioned equality operator or not "==", comparing String with equality operator is common mistake which works in some case and doesn't work in other. next String interview question is follow-up up of this.

Can we compare String using == operator? What is risk?
As discussed in previous String question, You can compare String using equality operator but that is not suggested or advised because equality operator is used to compare primitives and equals() method should be used to compare objects. As we have seen in pitfall of autoboxing in Java that how equality operator can cause subtle issue while comparing primitive to Object, any way String is free from that issue because it doesn't have corresponding primitive type and not participate in autoboxing. Almost all the time comparing String means comparing contents of String i.e. characters and equals() method is used to perform character based comparison. equals() return true if two String points to same object or two String has same contents while == operator returns true if two String object points to same object but return false if two different String object contains same contents. That explains why sometime it works and sometime it doesn't. In short always use equals method in Java to check equality of two String object.

How substring method work in Java?
This is one of the tricky Java question relate to String and until you are familiar with internals of String class, its difficult to answer. Substring shares same character array as original String which can create memory leak if original String is quite big and not required to retain in memory but unintentionally retained by substring which is very small in size and prevents large array from begin claimed during Garbage collection in Java. See How Substring works in Java for more details.

What is String pool in Java?
Another tough Java question asked in String interview. String pool is a special storage area in Java heap, mostly located on PerGen space, to store String literals like "abc". When Java program creates a new String using String literal, JVM checks for that String in pool and if String literal is already present in pool than same object is returned instead of creating a whole new object. String pool check is only performed when you create String as literal, if you create String using new() operator, a new String object will be created even if String with same content is available in pool.

What does intern() method do in Java?
As discussed in previous String interview question, String object crated by new() operator is by default not added in String pool as opposed to String literal. intern() method allows to put an String object into pool.