Strings

1. What is a String?
String is a series of characters, which is enclosed within double quotes and is treated as an object.

2. Is String an object in Java?
Yes

3. Can we access individual character in String?
NO

4. Does a Java String end with a null character?
No

5. What are the types of String in Java?
Mutable, immutable

6. Who de-allocates memory of a string?
Garbage-collector

7. What are the two ways of creating immutable strings?
String s=”Rama”;
String s=new String(“Rama”);

8. Why are immutable strings required?
In real world immutable String data exists: dob, gender, name, father name ,mother name .

9. How are mutable strings created in Java?
StringBuffer and StringBuilder

10. Why are mutable strings required in Java?
Real life has mutable string data : Address, qualification, password

11. Where are immutable strings created in Java?
heap Segment

12. What is a constant pool?

Is a region of the heap segment on the ram where memory is allocated for such strings that are created without using the new operator. Duplicates are not permitted in this region.

13. What is non constant pool?

Is a region of the heap segment where memory is allocated for such strings which are created using the new operator or using expressions. Duplicates are permitted in this region.

14. What is the role of equals() in Java?

It compares two strings

15. What is the role of == operator in Java?

It compares two references

16. What is the role = operator in Java?

Assignment operators are used in **Java** to **assign** values to variables. For example, int age; age = 5; The **assignment operator** assigns the value on its right to the variable on its left, and hence it is right to left associative.

17. Name some of the commonly used in built methods of String class?

startsWith(),endsWith(),toUpperCase(),toLowerCase(),contains().,etc.

18. Mention the difference between C Strings and Java Strings.

C-String	Java-String
null terminated,	not null terminated
not treated as an object	treated as an object
Memory is not allocated on the heap seg	is allocated in the heap segment
Memory is not deallocated by the garbage collector	it is deallocated by the garbage collector

individual characters in the string can be accessed	individual characters in the string cannot be accessed by the programmer's
not categorization as immutable and mutable	Categorized as immutable and mutable.

19. How do you compare two Strings in Java?

equals(), compareTo() , ==, equalsIgnoreCase()

20. What is a String pool in Java?

Where memory is allocated for strings, it is divided into 2 regions namely **constant pool** and **non-constant pool.**

21. Can we use String in switch case?

Upto jdk 1.7, only integers were allowed. From jdk 1.8 onwards even Strings are permitted in switch case.(Ex.Refer class notes.)

22. Why is String class considered immutable?

Strings which are created using "String" class cannot be altered.Hence String class is considered as immutable.

23. How is it possible for two string objects with identical values not to be equal under == operator?

i) If one String is in constant pool and the other String is in non-constant pool
ii) If both Strings are present in non constant pool.

24. Explain the difference between the following statements?

String s="Welcome to ABC";
The above string would be created in constant pool region.

String s= new String("Welcome to ABC");
The above String would be created in the non-constant pool region.

25. How to concatenate two different Strings?

Strings can be concatenated in following three ways
i) Using concat()
i) Using append()
i) "+" operator.

26. What is the output of the following program?

```
class Test
{
    public static void main(String[]  args)
    {
            String s="Hello";
            String s1=new String("Hello");
            System.out.println("s==s1 "+(s==s1));
            System.out.println("s.equals(s1)"+(s.equals(s1));
    }
}
```

Ans: s==s1 false
s.equals(s1) true

27. Is null a keyword in java?

No. Null is not a keyword. Rather it is a literal with special meaning. Null is the value of a reference variable. However we cannot use the word "null" to create a variable.

Eg. String s = null; valid statement.
String null; invalid statement.

28. What happens when you add a char value to a String?

The char type data would be converted to a String and would be concatenated with the existing String.

Eg:

```
System.out.println("Hello  world!" +A);
System.out.println("Hello  world!" +999);
System.out.println("Hello  world!" +999.9);
System.out.println("Hello  world!" +999.9f);
System.out.println("Hello  world!" +true);
```

Output:

Hello world! A
Hello world! 999
Hello world! 999.9
Hello world! 999.9
Hello world! true

Note:

System.out.println can only print String. If we try to concatenate character integer, float, boolean etc., then System.out.println automatically converts those values to strings using toString() and valueOf() methods.

System.out.println("Hello world!" +999);

System.out.println("Hello world!" +999.toString());
would be carried out internally.

System.out.println("Hello world!" +999.valueOf());

System.out.println("Hello world!" +"999.9");

System.out.println("Hello world! 999");

Output:
Hello world! 999.

Note:
System.out.println cannot print any other type of data. If some other type of data is given to System.out.println, then immediately toString() method would be called automatically. This inturn calls valueOf() method.

The valueOf() method gives string equivalent of the new data. It is this string equivalent that gets printed.

Eg:
int a =999;

```
System.out.println(a);

System.out.println(a.toString());

System.out.println(a.valueOf());

System.out.println("999");
```

Output:
999(which is actually a string).

The char, int, double, float, Boolean type data would be converted with the existing string and would be printed.

29. What happens when you add a int value to a String?

Refer Q.28.

30. What happens when you add a double value to a String?

Refer to Q.28.

31. What happens when you add a float value to a String?

Refer to Q.28.

32. What happens when you add a Boolean value to a String?

Refer to Q.28.

33. Can we directly print objects?

No. Instead of the object value getting printed , Object address gets printed as shown below

```
class Dog
{
    String name;
    String breed;
    int cost;
}
```

```
class Launch
{
    public static void main(String  args[])
    {
            Dog d=new Dog();
            d.name="Labo";
            d.breed="BullDog";
            d.cost=9000;
            System.out.println(d);//Directly  printing  the object
    }
}
```

output:Dog@ 15db9742

34. Can we directly print primitive data types?

Yes.

Eg: int a=99;

System.out.println(a);

àOutput:99

35. Does ArrayIndexOutOfBoundsException occur in case of immutable Strings? Why?

No. This problem doesnot occur because individual characters cannot be accessed in Java Strings.

36. Does ArrayIndexOutOfBoundsException occur in case of mutable Strings? Why?

No. This problem doesnot occur because individual characters cannot be accessed in Java Strings.

37. Name a few Immutable classes?

Immutable class means that once the String **object** is created its value cannot be altered. In Java, all the wrapper classes such as Byte, Short,Integer,Long,Char,Float,Double,Boolean and String class are immutable.

38. Is the String class mutable or immutable?

Immutable

39. In which package is the String class present?

java.lang package

40. What is a String literal?

Any value which is enclosed within double quotes is called as string literal.

41. What are the four ways of creating a String object?

1) String s="rama";
2) String s=new String("rama");
3) char name[]={'r','a','m','a');
   String s=new String(name);

4) char[] name={'r','a','m','a');
   String s= new String(name);

42. What are the two ways in which Strings can be concatenated?

Strings can be concatenated in following three ways

i) Using concat()
ii) Using append()
iii) "+" operator.

43. In how many ways can String comparison be performed?

There are four ways to compare two Strings in java.

i) Using equals() which compares two Strings.

ii) Using equalsIgnoreCase() which compares two strings by ignoring the case sensitivity.

iii) Using compareTo() which compares two strings character by character.

iv) Using == which compares references of two strings.

44. Give the memory map for String s1= "Rama";

Stack segment **Heap Segment**

2000

CP

2000

R A M A

S1

NCP

45. What is the role of charAt()?
It returns the character at specific index position.

46. What is the role of equals()?
It compares two strings by considering case sensitivity.

47. What is the role of equalsIgnoreCase()?
It compares two strings by ignoring case sensitivity.

48. What is the role of length()?
It is used to return the length of the string.

49. What is the role of substring()?

substring() is used for getting a substring of a particular String.
(example: refer class notes)

50. Explain the two versions of substring()?

String s = "RajaRamMohanRoy";

There are two variants of substring() method:

1) String substring(int beginIndex):
s.substring(7);
The above statement would display a substring from 7th character onwards till the end of the String. (MohanRoy)

2) String substring(int beginIndex, int endIndex):
s.substring(7,12);
The above statement would display a substring from 7th character to 11th character. (Mohan)

51. What is the role of toLowerCase()?

It returns a string by converting it into lower case.

52. What is the role of toUpperCase()?

It returns a string by converting it into upper case.

53. What is the role of valueOf()?

valueOf() method converts different types of values into String. Using valueOf() method we can convert int to string, char to string, float to string, Boolean to string etc., It returns string representation of the given value.

54. How does valueOf() behave on objects?

It returns the string representation of an object.

55. What is the role of toString()?

toString() is used to convert a non-string object to string object.

56. What is the role of contains()?

It is used to verify that if a string contains a particular substring or not.
Ex: refer class notes.

57. What is the role of concat()?

The concat() method concatenates is used to concatenate or combine

two strings.

58. What is the role of isEmpty()?

It is used to check whether the string is empty or not.

59. What is the role of indexOf()?

It returns the index of a sprcified character within a string

60. Which classes are used to create mutable Strings in java?

StringBuffer and StringBuilder builtin classes would be used to create mutable strings in java.

61. What is the difference between StringBuffer and StringBuilder class?

(Refer class notes)

62. What are the three constructors of StringBuffer class?

StringBuffer s=new StringBuffer("RAMA");
StringBuffer s=new StringBuffer();
StringBuffer s=new StringBuffer(10);

63. What is the default buffer size of the StringBuffer class?

The initial capacity of the **StringBuffer class** is 16.

64. What is the role of append()?

It is used to concatenate or combine two mutable strings.

65. What is the role of capacity()?

It returns the capacity of StringBuffer or StringBuilder.

66. What is the role of ensureCapacity()?

It is used to change the capacity of StringBuffer or StringBuilder.

67. What are the three constructors of StringBuilder?

StringBuilder s=new StringBuilder("RAMA");
StringBuilder s=new StringBuilder();
StringBuilder s=new StringBuilder(10);

68. In which class is the toString() present?

Object class

69. What is the role of intern()?

It is used to create a copy of the string present in non-constant pool into the constant pool region.

70. What is the role of StringTokenizer class?

It is used to chop a large string at a specified character in order to provide tokens.

Eg:

```
import java.util.*;
class Abc
{
    public static void main(String args[])
    {
        StringTokenizer st=new StringTokenizer("ABC for technology training");
        while(st.hasMoreTokens()==true)
        {
            System.out.println(st.nextToken());
        }

    }

}
```

Output: ABC
for
technology
training

71. In which package is the StringTokenizer class present?

java.util

72. When was the StringBuilder class introduced in Java?

jdk 1.5

73. Can we explicitly put a string object which is in the non-constant pool into the constant pool?

Yes, by using intern(). But for the reverse operation there is no method.

74. What is an immutable object?

An object whose state can never be changed is called an immutable object.

75. What would happen if an immutable object is attempted for modification?

A copy of the object would be taken and would be modified whereas the original object would still remains immutable.

76. Which exception is generated if we try to mutate an immutable object?

IllegalObjectStateException

77. Where does the String pool exist?

String pool exists on the perm space on the heap segment.

78. Is String class final in Java?

Yes.

79. Why is char array preferred to store password in Java than a String?

Passwords are normally frequently modified. Hence it must be created either using **StringBuffer** or **StringBuilder** class or **char** array can be used.

80. Why is the String class final in Java?

Because String class is immutable, it has made as final class so that inheritance of the String class can be prevented.

81. Is the String class thread safe?

Yes

82. Is StringBuffer class thread safe?

Yes

83. Is StringBuilder class thread safe?

No

84. What is the difference between declaring a String and defining a String?

Declaration

String s;

Defining

String s = “Rama”;