String Questions

1 .String reverse using recursion

1. public class ReverseString {
2. public static void main(String[] args) {
3. String myStr = "Guru99";
4. //create Method and pass and input parameter string
5. String reversed = reverseString(myStr);
6. System.out.println("The reversed string is: " + reversed);
7. }
8. //Method take string parameter and check string is empty or not
9. public static String reverseString(String myStr)
10. {
11. if (myStr.isEmpty()){
12. System.out.println("String in now Empty");
13. return myStr;
14. }
15. //Calling Function Recursively
16. System.out.println("String to be passed in Recursive Function: "+myStr.substring(1));
17. return reverseString(myStr.substring(1)) + myStr.charAt(0);
18. }
19. }

2. How to remove blank space from string

public class Test1 {

public static void main (String[]args)

{

String a =null;

Scanner scan = new Scanner(System.in);

System.out.println("\*\*\*\*\*\*\*\*\*White Space Remover Program\*\*\*\*\*\*\*\*\*\*\*\*\n");

System.out.println("Enter your string\n");

a = scan.nextLine();

System.out.println("Input String is :\n"+a);

String b= a.replaceAll("\\s+","");

System.out.println("\nOutput String is :\n"+b);

}

}

3.String is not primitive it is derived data type.

// Java program to print all permutations of a

// given string.

public class Permutation

{

    public static void main(String[] args)

    {

        String str = "ABC";

        int n = str.length();

        Permutation permutation = new Permutation();

        permutation.permute(str, 0, n-1);

    }

    /\*\*

     \* permutation function

     \* @param str string to calculate permutation for

     \* @param l starting index

     \* @param r end index

     \*/

    private void permute(String str, int l, int r)

    {

        if (l == r)

            System.out.println(str);

        else

        {

            for (int i = l; i <= r; i++)

            {

                str = swap(str,l,i);

                permute(str, l+1, r);

                str = swap(str,l,i);

            }

        }

    }

    /\*\*

     \* Swap Characters at position

     \* @param a string value

     \* @param i position 1

     \* @param j position 2

     \* @return swapped string

     \*/

    public String swap(String a, int i, int j)

    {

        char temp;

        char[] charArray = a.toCharArray();

        temp = charArray[i] ;

        charArray[i] = charArray[j];

        charArray[j] = temp;

        return String.valueOf(charArray);

    }

}

4. What is the output of this program?

1. **class** String\_demo
2. {
3. **public** **static** **void** main(String args[])
4. {
5. **char** chars[] = {'a', 'b', 'c'};
6. String s = **new** String(chars);
7. System.out.println(s);
8. **int** ascii[] = { 65, 66, 67, 68};
9. String s = **new** String(ascii, 1, 3);
10. System.out.println(s);
11. }
12. }

Ans---------abc

Bcd

5.What is output

String s=new String("Hello");

String s1=new String("Hellow");

System.out.println(s=s1); ans------------Hellow

6.StringBuffer s=”komal”;

String s1=”komal”;

Syso(s==s1);//false

Syso(s.equals(s1))//false

7.Permutations in string

8.String rotation

// Java program to check if two given strings are

//  rotations of each other

class StringRotation {

    /\* Function checks if passed strings (str1 and str2)

       are rotations of each other \*/

    static boolean areRotations(String str1, String str2)

    {

        // There lengths must be same and str2 must be

        // a substring of str1 concatenated with str1.

        return (str1.length() == str2.length()) &&

               ((str1 + str1).contains(str2));

    }

    // Driver method

    public static void main(String[] args)

    {

        String str1 = "AACD";

        String str2 = "ACDA";

        if (areRotations(str1, str2))

            System.out.println("Yes");

        else

            System.out.printf("No");

    }

}

9.how String is different in c and java

In C, a string is typically just an array of (or a pointer to) chars, terminated with a NUL (\0) character. You can process a string as you would process any array.

In Java, however, strings are not arrays. Java strings are instances (objects) of the java.lang.String class. They represent character data, but the internal implementation is not exposed to the programmer. You cannot treat them as arrays, although, if required, you can extract string data as an array of bytes or chars (methods getBytes and getChars). Note also that Java chars are 16-bits, always, while chars in C are typically (not always) 8-bit.

**10)  When you execute String str = new String("abcd")? how many String objects are created?**([answer](http://javarevisited.blogspot.sg/2015/10/133-java-interview-questions-answers-from-last-5-years.html))  
This is another tricky Java interview question as many Java developer will answer just one but that's not true. There are two String objects are created here, the first String object is created by String literal "abcd" and the second one is created by new String().  If you are not sure about how  
  
  
**. Can We Use String In Switch Case Statement?**

**Answer :**

Yes from Java 7 string can be used in switch case statement.

**Q5.  what will be the output of this code ?**  
 **public static void main(String[] args)**  
 **{**  
 **StringBuffer s1=new StringBuffer("Buggy");**  
  
**test(s1);**  
  
 **System.out.println(s1);**  
  
 **}**  
  
 **private static void test(StringBuffer s){**  
 **s.append("Bread");**  
 **}**  
  
Ans. BuggyBread  
  
**Q8.  what will be the output of this code ?**  
 **public static void main(String[] args)**  
 **{**  
 **String s1=new String("Buggy");**  
  
**test(s1);**  
  
 **System.out.println(s1);**  
  
 **}**  
  
 **private static void test(StringBuffer s){ //compiler error**  
 **s.append("Bread");**  
 **}**  
  
Ans. Buggy

If all StringBuffer then it will print BuggyBread  
  
**Q9.  what will be the output of this code ?**  
 **public static void main(String[] args)**  
 **{**  
 **StringBuffer s1=new StringBuffer("Buggy");**  
  
**test(s1);**  
  
 **System.out.println(s1);**  
  
 **}**  
  
 **private static void test(StringBuffer s){**  
 **s=new StringBuffer("Bread");**  
 **}**  
  
Ans. Buggy

10. public static void main(String[] args)

{String str="komal";

char c[]=new char[10];

str.getChars(0,4,c,0);

System.out.println(c);

}

Ans----koma

1. why string is immutable?//Security n memory enhancement
2. how do we check string contains digit only?

String string = "-1234.15";  
        boolean numeric = true;  
  
        numeric = string.matches("\\d+");

Or

System.out.println(str1 + ":" + str1.matches("[0-9]+"));

|  |
| --- |
| **Question: How does substring() method works on a string?** |
| |  | | --- | | **Answer:** | | String in java is a sequence of characters. String is more like a utility class which works on that character sequence. This character sequence is maintained as a array called value[], for example  *private final char value[];*  String internally defines two private variables called offset and count to manage the char array. The declarations can be as shown below:  */\*\* The offset is the first index of the storage that is used. \*/ private final int offset;  /\*\* The count is the number of characters in the String. \*/ private final int count;*  Everytime we create a substring from any string object, substring() method assigns the new values of offset and count variables. The internal char array is unchanged. This is a possible source of memory leak if substring() method is used without care. | |

1. How does garbage collector know string is not in used need to removed?
2. public static void main(String []args){

String str="JavaProgramming";

String str1="java";

System.out.println(str.compareTo(str1)); -32 in case of lowercase 11 and -11

System.out.println(str1.compareTo(str)); 32

}

32