1. Write a program to reverse a String :

For example String s= “abcdef” expected Output --- > fedcba

Ans:

```
 public static void main(String[] args) {
                    String s = "abcdef";
                    String rev="";

                    for (int i=(s.length()-1); i>=0; i--)
                    {
                              rev=rev+s.charAt(i);

                    }
                    System.out.println("The reversed String is "+rev);


          }
```

2. Write a program to add the integers available in the string:

For example : String s = “10value8with20value”; then the sum should be10+8+20 = 38

Ans:

```
          static String s = "60fdf5ffrf80frfr4fr5";

          public static void main(String[] args) {
                    String number = "";
                    int temp = 0;
                    int flag = 0;
                    for (int i = 0; i < s.length(); i++) {
                              if (Character.isDigit(s.charAt(i))) {

                                        number = number + s.charAt(i);

                                        flag = 1;

                                        if (i != (s.length() - 1))
                                                  continue;

                              }

                              if (flag == 1) {

                                        int value = Integer.parseInt(number);
                                        number = "";
                                        temp = value + temp;

                                        flag = 0;
                    }

                }

                System.out.println("the addition of number are :" + temp);
        }
```

3. WAP to count the number of occurrence of a single character in a String:

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

                String s = "abcdefabcdef";
                int count = 0;

                for (int i=0; i<s.length(); i++)
                {
                        if (s.charAt(i)=='a')// to count the occurrance of character 'a' in the string
                        {
                                count = count +1;
                        }

                }

                System.out.println("The character a is available for "+count+ " times");


}
```

4. WAP to count the number of occurrence of characters in a String:

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

Alternative approach using HashMap

```
public static void main(String[] args) {
                String s = "abcdefabcdefaabb";

                HashMap<Character, Integer> hm = new HashMap<Character,
Integer>();

                for (int i=0; i<s.length(); i++)
                {

                        char charvalue = s.charAt(i);//a
```

```
                    if(hm.containsKey(charvalue))
                    {
                            hm.put(charvalue, hm.get(charvalue)+1);
                    }
                    else
                    {
                            hm.put(charvalue, 1);

                    }
            }

            System.out.println(hm);


    }
```

5. WAP to count a pattern to be available in a given String

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

String pat = “abc”;
String txt = “abcdefabcdef”;
 int M = pat.length();
      int N = txt.length();
      int res = 0;

      /* A loop to slide pat[] one by one */
      for (int i = 0; i <= N - M; i++) {
         /* For current index i, check for
      pattern match */
         int j;
         for (j = 0; j < M; j++) {
            if (txt.charAt(i + j) != pat.charAt(j)) {
               break;
            }
         }

         // if pat[0...M-1] = txt[i, i+1, ...i+M-1]
         if (j == M) {
                res++;
                j = 0;
            }
        }
        System.out.println("the count is :"+res);
    }
```

OR

```
public static void main(String[] args) {
                String s = "ghijalalalalalalalalkldjalsaadkjkalsaghisjdalskdjaghi";
                int count = 0;
                Pattern p = Pattern.compile("lal");

        Matcher m = p.matcher(s);


        while(m.find())
        {
                count++;

        }

        System.out.println(count);


        }

}
```

6. WAP to remove duplicate characters from String:

```
    } public static void main(String[] args) {
                String s = "abcdefabcdef";
        String s2 = "";
        for (int i = 0; i < s.length(); i++) {
            Boolean found = false;
            for (int j = 0; j < s2.length(); j++) {
                if (s.charAt(i) == s2.charAt(j)) {
                    found = true;
                    break;
                }
            }
            if (found == false) {
              s2 = s2+ s.charAt(i);
            }
        }
        System.out.println(s2);
```

Alternate way is to be done through hashset:

7. WAP to remove the duplicate character from String and represent the character count next to it e.g. abcdefabcdef---→ expected output a2b2c2d2e2f2

```
public static void main(String[] args) {
            String s = "aaabdhhhssassa";
            String s2 = "J64446654ava8J";
            char[] c = s.toCharArray();
            int sz = c.length;
            int i = 0;
            String alphanumericalstring = "";
            int j = 0;
            int counter = 0;
            for (i = 0; i < sz; i++) {
                counter = 0;
                for (j = 0; j < sz; j++) {
                    if (j < i && c[i] == c[j]) {
                        break;
                    }
                    if (c[j] == c[i]) {
                        counter++;

                    }
                    if (j == sz - 1) {

                        String value = Integer.toString(counter);

                        String modifiedstring = c[i] + value;

            alphanumericalstring = alphanumericalstring + modifiedstring;
                }
```

```
                        }

                }
                System.out.println(alphanumericalstring);

        }
```

OR

```
public static void main(String[] args) {
                String s = "asjchgjqdksdhsdkasassaa";

                HashMap<Character, Integer> hm = new HashMap<Character,
Integer>();

                for(int i=0; i<s.length(); i++)
                {
                        char charvalue = s.charAt(i);


                        if(hm.containsKey(charvalue))
                        {
                                hm.put(charvalue, hm.get(charvalue)+1);

                        }

                        else
                        {
                                hm.put(charvalue, 1);
                        }


                }

                System.out.println(hm);


                Set<Entry<Character, Integer>> allkeyvalues = hm.entrySet();


                for(Entry<Character, Integer> e :allkeyvalues)
                {
```

```
                System.out.print(e.getKey()+ " "+e.getValue()+" ");
        }



    }
```

8. WAP to reverse the complete sentence for example String s = “This is String” then the expected output should be Reverse string = “String is This”

```
public static void main(String[] args) {
        String s = "This is String";
        String rev="";

        String[] Splitvalue = s.split(" ");

        int size = Splitvalue.length-1;

        for(int i =size; i>=0; i--)
        {
            System.out.print(Splitvalue[i]+" ");
        }



    }
```

9. WAP to remove the alphabets from the String

Ans.
```
public static void main(String[] args) {
        String s = "d5de5dd56d5dd";
        String numericstring = "";

        for (int i = 0; i < s.length(); i++) {
            if (Character.isDigit(s.charAt(i))) {
                numericstring = numericstring + s.charAt(i);

            }

        }
        System.out.println("The numeric string is :" + numericstring);
```

```
            }
```

10. WAP to remove the numbers from the String   no chi string nahi anachi ahe o/p madhe

```
public static void main(String[] args) {
                        String s = "d5de5dd56d5dd";
                        String numericstring = "";

                        for (int i = 0; i < s.length(); i++) {
                                    if (!Character.isDigit(s.charAt(i))) {
                                                numericstring = numericstring + s.charAt(i);

                                    }

                        }
                        System.out.println("The numeric string is :" + numericstring);

            }
```

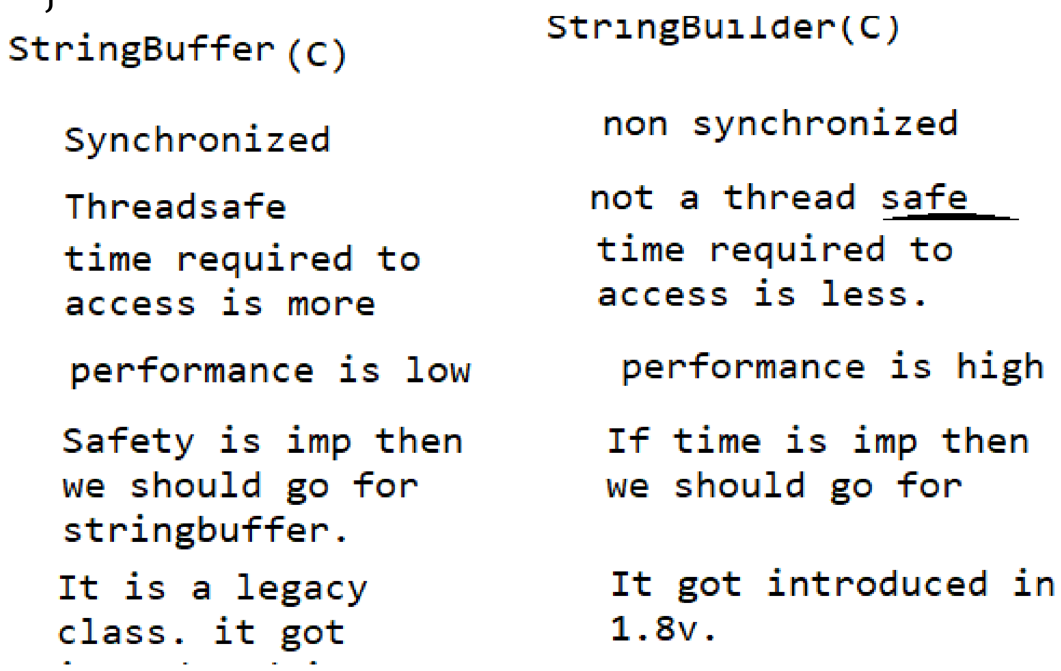
11. WAP to add all the number individually from the String for example if the string is "ab5ds51s2" then output should be 5+5+1+2 = 13

```
Ans. public static void main(String[] args) {
                        String s = "ab5ds51s2";
                        int digit = 0;
                        String numericstring = "";
                        for (int i = 0; i < s.length(); i++) {
                                    if (Character.isDigit(s.charAt(i))) {
                                    numericstring=  s.substring(i, i+1);

                                    digit = digit+ Integer.parseInt(numericstring);

                                    }
                        }
                        System.out.println("The numeric string is :" + digit);
            }
```

Alternate way for this:

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

                        String s = "45sadasd7sdsa12sdsads8";
                        String num = "";


                        int temp =0;

                        for(int i =0; i<s.length(); i++)
                        {
                                    if(Character.isDigit(s.charAt(i)))
```

```
                {
                        num = num+ s.charAt(i);
                        int intvalue = Integer.parseInt(num);
                        temp= temp +intvalue;
                        num = "";
                }

        }

        System.out.println("The sum is :"+temp);


    }
```



StringBuffer (C)	StringBuilder(C)
Synchronized	non synchronized
Threadsafe	not a thread safe
time required to access is more	time required to access is less.
performance is low	performance is high
Safety is imp then we should go for stringbuffer.	If time is imp then we should go for
It is a legacy class. it got	It got introduced in 1.8v.