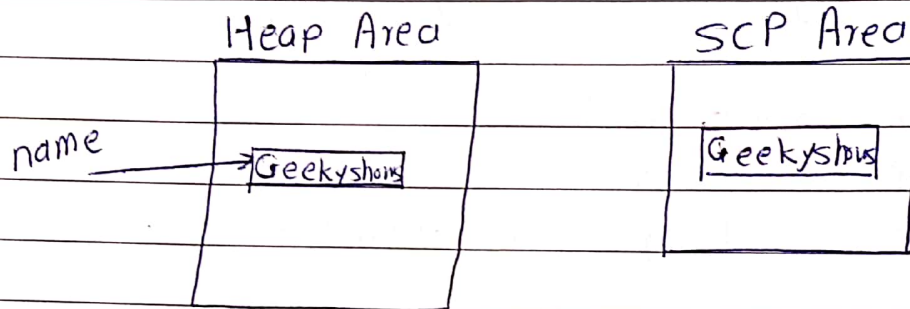
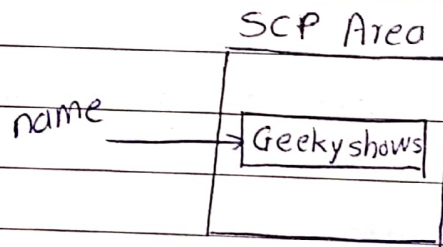


Q.1 Difference bet<sup>n</sup> String s = "Hello" & String s = new String("Hello")

Ans String name = new String("Geekyshows");  
In this two objects will be created one in heap area and the other object in String Constant Pool (SCP) and name is always pointing to heap object.



String name = "Geekyshows";  
Only one object will be created in SCP Area and name is pointing to that object.



Q.2 Difference bet<sup>n</sup> "==" and equals() method  
Basically in Java both "==" and equals() methods are used for comparing objects equality but only difference is that "==" is an operator and equals() is a method.

== operator

① "==" is an operator in Java

equals() method

① equals() is a method in Java.

2] "==" operator is used for reference comparison (address comparison)      ② equals() method is used for content comparison

③ It checks whether both objects points to the same memory location      ③ equals() method evaluates to the comparison of values in the object.

④ "==" is used to compare primitive eg. boolean, int, char, etc      ④ equals() method is used to compare objects in java eg. String object in Java

### Program for "==" operator

```
① Class Test
{
    psvm()
{
    String s1 = new String("Hello");
    String s2 = new String("Hello");
    sop(s1 == s2) // False

    String s3 = "amit";
    String s4 = "amit";
    sop(s3 == s4) ; // True
}
```

O/P      False

True



## ② Program for equals() method

Class Test

{

psvm()

{

String s1 = new String ("Hello");

String s2 = new String ("Hello");

sop ( s1.equals(s2)); // It will return true

sop ( s1 == s2) // It will return false

}

}

O/P

true

false

## ③ class Test {

psvm() {

String s1 = "Sachin"

String s2 = "Sachin";

String s3 = new String ("Sachin");

String s4 = "Saurav";

sop ( s1.equals(s2)); // True

sop ( s1.equals(s3)); // True

sop ( s1.equals(s4)); // False

}

}

## ④

String s1 = "Sachin";

String s2 = "Sachin";

String s3 = new String ("Sachin");

sop ( s1 == s2); // True [ because both refer same

sop ( s1 == s3); // False ( because s3 refers to instance created in nonpool

## Java program to count 'ch'

```

public class Demo {
    psvm()
    {
        String str = "The best";
        int count = 0;

        for (int i = 0; i < string.length(); i++)
        {
            if (string.charAt(i) != ' ')
            {
                count++;
            }
        }
        sop (" — " + count);
    }
}

o/p = 7
    
```



①

## Reverse Each Word of String

```
public class ReverseWord
{
    psvm()
    {
        String str = "My Name is Sita";
        String a[] = str.split(" ");

        for (int i = 0; i < a.length; i++)
        {
            sop(a[i] + " "); // print original string
        }
        sop("\n");
        for (int i = a.length - 1; i >= 0; i--)
        {
            sop(a[i] + " ");
        }
    }
}
```

O/P My Name is Sita  
Sita is Name My

②

## Reverse String

```
public class RevString {
    psvm()
    {
        String str = "Java";
        char a[] = str.toCharArray();

        for (int i = a.length - 1; i >= 0; i--)
        {
            sop(a[i]);
        }
    }
}
```

O/P avaJ

### ③ Java Program to remove Duplicate



Class IndexOf {

public static String unique (String s)

{

String str = new String ();

int len = s.length();

for (int i = 0; i < len; i++)

{

char c = s.charAt(i); // if c is present in str  
// it return index of c  
if (str.indexOf(c) < 0) else it return -1

{

str += c;

// adding c to str if -1  
is returned

}

}

return str;

}

psvm()

{

String s = "abbc";

sop(unique(s));

}

}

O/P abc

### ④ How to Extract digits from String

String str = "The Price of the Book 49";

str, no = str.replaceAll("[^0-9]", "");

sop(no);

O/P 49



# Java Program To Reverse

## Individual Each Word of String.



```
public class ReverseEachWord {  
    psvm()
```

```
{
```

```
    String str = "Welcome To Java";
```

```
    String[] words = str.split(" ");
```

```
    String reverseString = "";
```

```
    for (String w : words)
```

```
    {
```

```
        String reverseWord = "";
```

```
        for (int i = w.length() - 1; i >= 0; i--)
```

```
        {
```

```
            reverseWord = reverseWord + w.charAt(i);
```

```
        }
```

```
        reverseString = reverseString + reverseWord + " ";
```

```
    }
```

```
    sop(reverseString);
```

```
}
```

```
}
```

O/P

ava oT hcleme

50 Convert every word of a String to Uppercase.



```
class Main {
```

```
    psvm()
```

```
{
```

```
    String message = "everyone loves Java";
```

```
    char[] charArray = message.toCharArray();
```

```
    boolean foundSpace = true;
```

```
    for(int i = 0; i < charArray.length; i++)
```

```
    {
```

```
        // if the array element is a letter
```

```
        if (Character.isLetter(charArray[i])) {
```

```
            // check if space present before letter
```

```
                if (foundSpace)
```

```
                {
```

```
                    charArray[i] = Character.toUpperCase  
                        (charArray[i]);
```

```
                    foundSpace = false;
```

```
                }
```

```
            }
```

```
            else {
```

```
                // if the new character is not character
```

```
                foundSpace = true;
```

```
            }
```

```
        }
```

```
        // convert the char array to String
```

```
        message = String.valueOf(charArray);
```

```
        sop("Message" + message);
```

```
    }
```

```
}
```

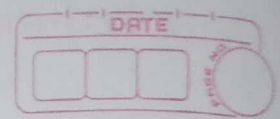
O/P

Everyone Loves Java



1Q

## Java Program To Find Duplicate character in a Given String



```
public class DuplicateString {  
    public static void main(String args[])  
    {  
        String string1 = "abcde bcd";  
        int count;
```

```
//convert given string into character array  
char string[] = string1.toCharArray();
```

```
sop("Duplicate characters in a given string.");
```

```
//count each character present in the string
```

```
for(int i = 0; i < string.length; i++) {  
    count = 1;
```

```
for(int j = i+1; j < string.length; j++) {
```

```
if(string[i] == string[j] && string[i] != '\0') {
```

```
    count ++;
```

```
//Set string[j] to 0 to avoid printing visited character
```

```
    string[j] = '0';
```

```
}
```

```
}
```

```
// A character is considered as duplicate if  
count is greater than 1
```

```
if(count > 1 && string[i] != '0')
```

```
    sop(string[i]);
```

```
}
```

O/P

bc