j==0 j==n/2 j==n-1

---

Array:-

| Variable | Array |
|---|---|
| 1.Using variable we can store only one value.<br>Ex:<br>int x=20;<br>  | 1.We can store multiple value using array.<br>  |



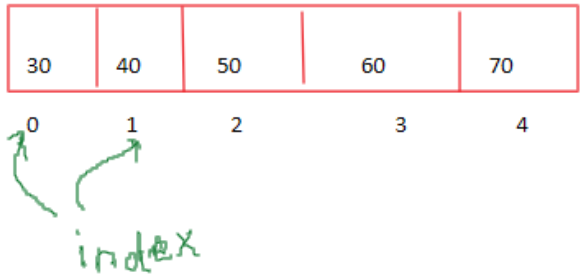


Basic syntax:-
datatype[] arrayname={val1,val2,val3,val4};
int[] ar={10,20,30,40};
int []ar={10,20,30,40};
int ar[]={10,20,30,40};

Array:-Array is a container which is used to store similar type of values.

Array:-Array is a container which is used to store similar type of values.
Array is collection of homogenious values.

Int[] ar={30,40,50,60,70};

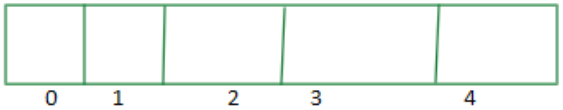



-All values will be stored based on index value in array.
-Array index value always start from zero.
-We find the size of array by using length variable.
Eg: ar.length
ArrayIndexOutOfBoundsException:-Whenever we try to fetch data from array by using index which index value is not present then we get ArrayIndexOutOfBoundsException.
Ex:-
Int[] ar={30,40,50,60,70};
Sopl(ar[5]);//ArrayIndexOutOfBoundsException

#By using new keyword syntax:-
Syntax:- Datatype[] arrayname=new datatype[size];//providing size is mandatory.
Ex:-int[] ar=new int[5];

present then we get ArrayIndexOutOfBoundsException.
Ex:-
Int[] ar={30,40,50,60,70};
Sopl(ar[5]);//ArrayIndexOutOfBoundsException

#By using new keyword syntax:-
Syntax:- Datatype[] arrayname=new datatype[size];//providing size is mandatory.
Ex:-int[] ar=new int[5];




To initialize array value:
Syntax-arrayname[index]=value;
Ex:-
String[] names=new String[5];
names[0]="virat";
names[1]="Dhoni";
names[2]="Sachin";
names[3]="Rahul";
names[4]="Sehwag";
Disadvantages of an array:-
-Array size is fixed we cannot change the size of an array during runtime.
-Array is homogeneous.
-In array there is no inbuilt methods to deal with data.

---

String

Inbuilt class

java.lang

String-> String is an inbuilt class which is present in java.lang package.

String class contains constructors and methods.

String is final class we cannot inherit or extend it.

Two ways to create a string object:-

1)By using assignment(=) operator

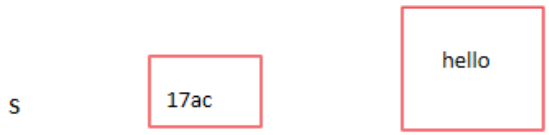
Ex:-String s="Hello";

2)By using new Keyword.

---

1)String s="hello";

Sopl(s);//hello

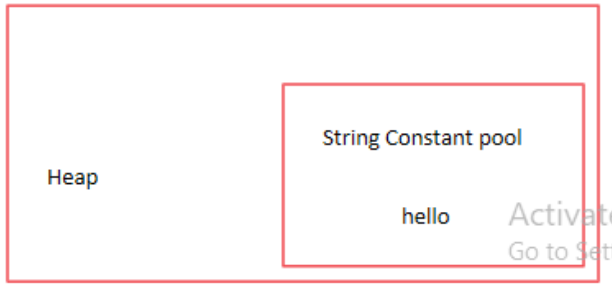



Here s prints the value not the address

String Constant pool-

Scp is a memory which is used to store String object.

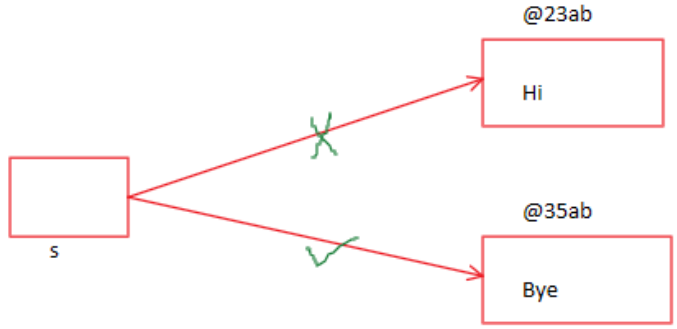
Scp is part of heap memory.



Inside String class toString() is already overriden hence we try to print object refrence it will always print the content of the object.

---

Ex-
String s="Hi";
Sopl(s);
s="Bye";
Sopl(s);




String is immutable, once we create a String Object we cannot change the value of string.
Whenever we try to change the string value internally one new object will be created.
String is a final class.
Q.Why String is immutable?

---

Ex:
String s="JavaProgram";
String s1="JavaProgram";

Whenever we try to change the string value internally one new object will be created.
String is a final class.

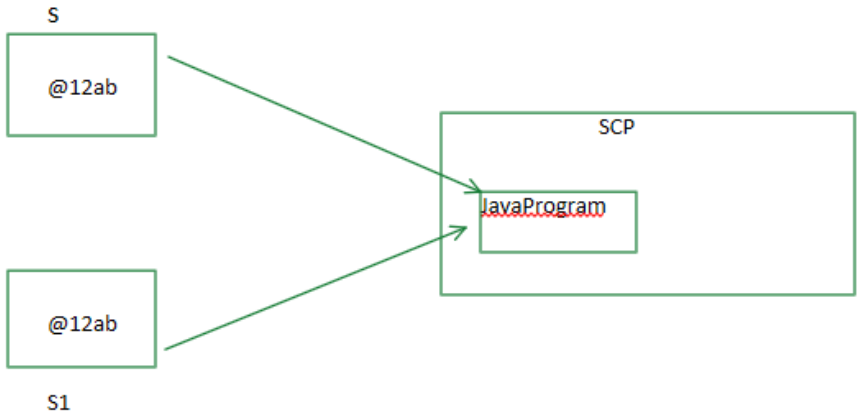
Q.Why String is immutable?

---

Ex:

```
String s="JavaProgram";
String s1="JavaProgram";
Sopl(s==s1);//true
```

address compares

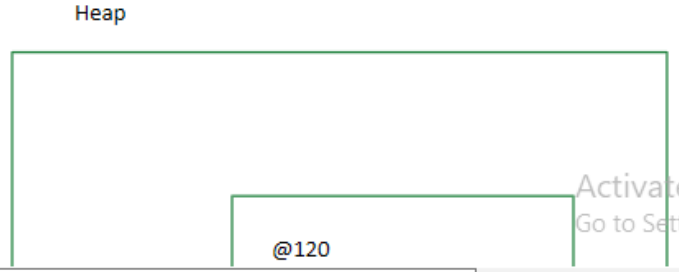



Inside String Constant pool duplicate values are not allowed.
If we try to store duplicate values then it will refer to the existing object instead of creating new object.

---

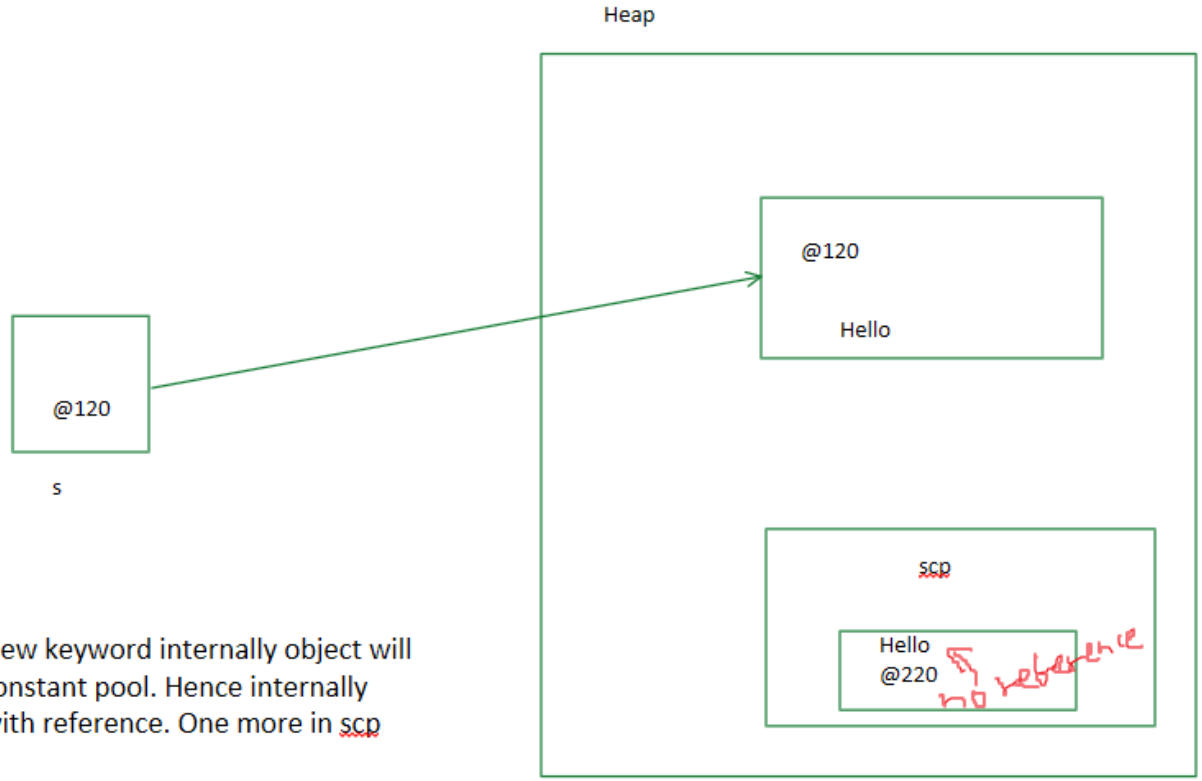
Create String by using new keyword:

```
String s=new String("Hello");
Sopl(s);
```



Create String by using new keyword:

```
String s=new String("Hello");
Sopl(s);
```




Whenever we create a String object by using new keyword internally object will be created inside heap area as well as String constant pool. Hence internally two objects will be created one in heap area with reference. One more in scp without reference.

---

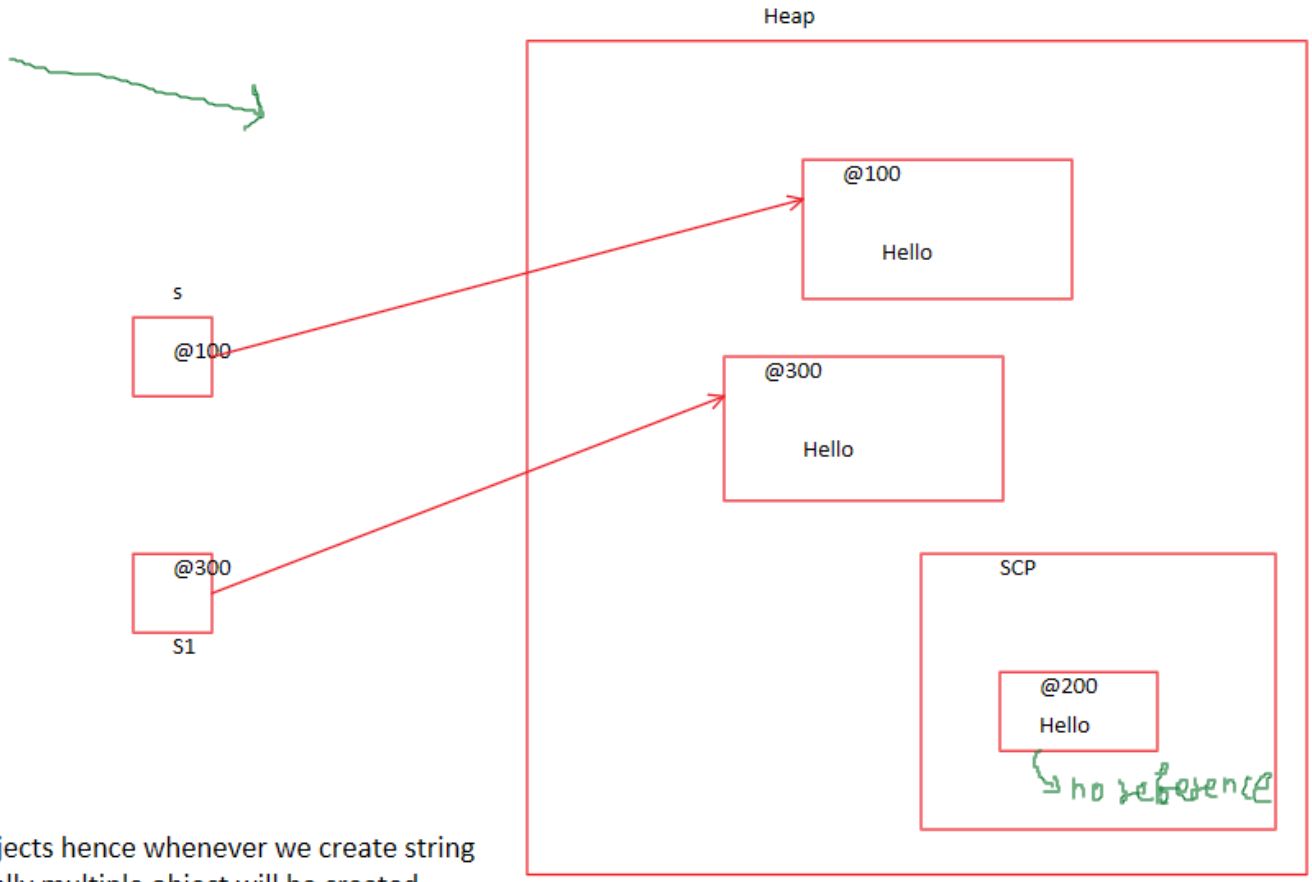
Ex:

```
String s=new String("Hello");
```

without reference.

---

Ex:
String s=new String("Hello");
String s1=new String("Hello");
Sopl(s==s1);//false




Heap area allows duplicate objects hence whenever we create string object with same value internally multiple object will be created with different address.
Note-Inside string class equals() is already overriden.

Heap area allows duplicate objects hence whenever we create string object with same value internally multiple object will be created with different address.
Note-Inside string class equals() is already overriden.
Ex:

```
String s=new String("Hello");
String s1=new String("Hello");
Sopl(s==s1);//false
Sopl(s.equals(s1));//true
```

Difference between == and equals()
1)== is operator while equals() is method.
2)== compares the address of object but equals() compares the content of the object.

Methods of String class:-
1)charAt()-
charAt() is a non static method of string class its taking int index as argument and return type is char.

```
Ex:     String s="Java";
      char ch=s.charAt(0);
      System.out.println(ch);//J
```

2)toUpperCase():-
toUpperCase() is a non static method of string class which is used to convert lower case alphabet into uppercase.
String is return type of toUpperCase().

```
System.out.println(ch);//J
```

2)toUpperCase():-

toUpperCase() is a non static method of string class which is used to convert lower case alphabet into uppercase.

String is return type of toUpperCase().

Ex:

```
String s="Java";
String s1=s.toUpperCase();
System.out.println(s1);//JAVA
```

3)toLowerCase():-

toLowerCase() is a non static method of string class to convert uppercase to lowercase alphabets.

Ex:-

```
String s2="JAVA";
String s3=s2.toLowerCase();
System.out.println(s3);//java
```

4)length():-

Length() is a non static method of string class which is used to find the size of string.Return type of length() is int.

Ex:-

```
String s="Java";
System.out.println(s.length());//4
```

5.)trim()-

Trim() is a non static method of string class which is used to remove the unnecessary space of beginning of string and ending of string.

Ex:-

```
String s=" java is programming language ";
        String s1=s.trim();
        System.out.println(s);
        System.out.println(s1);
        o/p=
```

5.)trim()-
Trim() is a non static method of string class which is used to remove the unnecessary space of beginning of string and ending of string.
Ex:-

```
String s=" java is programming language ";
        String s1=s.trim();
        System.out.println(s);
        System.out.println(s1);
        o/p=
    java is programming language
  java is programming language
```

6)toCharArray()-
toCharArray() is a non static method of string class which is used to convert string into character array.
Return type of toCharArray() is char[].
Ex:

```
String s="java";
        char[] ch=s.toCharArray();
        for(int j=0;i<ch.length;j++)
        {
            System.out.println(ch[i]);
        }
```

O/p
j
a
v
a

7)substring()-
Public string substring(int beginning index, int ending index)
Public string substring(int beginning index)
Substring() takes beginning index but it not takes last index.
Ex:

```
String s="Meghalaya";
```

```
{                                                    a
    System.out.println(ch[i]);                       v
}                                                    a
```

7)substring()-
Public string substring(int beginning index, int ending index)
Public string substring(int beginning index)
Substring() takes beginning index but it not takes last index.
Ex:

```
String s="Meghalaya";
System.out.println(s.substring(4,7));//ala
```

8)split()-
Split(" ") is a non static method of string class which is used to convert string into small small string.
Return type of split() is string array.
Based on parameter split() will work.
If we give space as parameter up to space it will convert a separate string.
Ex:

```
String s="Honey good singer";
    String[] s1=s.split(" ");
    for (int j = 0; j < s1.length; j++) {
        System.out.println(s1[j]);
    }
```

O/p
Honey
good
singer

9)equalsIgnoreCase()-
equalsIgnoreCase() is used to compare two strings without considering cases(uppercase or lowercase).
Ex:

```
String s="uditraj";
String s1="UDITRAJ";
System.out.println(s.equals(s1));//false
System.out.println(s.equalsIgnoreCase(s1));//true
```