

Java.lang Package..

1-It is a very Most Important Package in Java.
Without this package u can not write or run single
java program.

2-It is default package.

This Package Provide us many Imp classes

A-Object class

b-String class

c-StringBuffer

d-StringBuilder

e-System class

d-wrapper classes

e-Exception Handling Related imp classes

f-Multithreading Related imp CLasses

g-Class class

e-Method class

Object Class:-

1-Object is Root class for All classes in Java

2-Every class in java is child class of Object class either directly or indirectly by default.

3-if our class extends any other class then it is indirectly child class of Object class through MultiLevel Inheritance(not multiple Inheritance).

4-All methods of Object class is available for All child classes in java.

11 -Methods of Object class

1-public String toString()

2- public native int hashCode()

3- public boolean equals(Object o)

4- protected void finalize()

5-protected native Objecte clone()

6-public final Class getClass()

7-public final void wait()

8-public final native void wait(long m)

9-public final void wait(long m,int n)

10-public native final void notify()

11- public native final void notifyAll()

12-private native void registerNatives()

Can not use in child classes..

1- `toString()`

This method is used to return address of an object.

In the form of `className@hexadecimalCODE`.

This method internally calls the `hashCode()`

method and convert it into Hexadecimal code.

Like this...

Public String `toString()`

{

Return

getClass().getName() + "@" + Integer.toHexString(hashCode())

}

2-`equals()`:= this method is used to compare the address of an objects. And return Boolean value.

Ex:-

```
Test t=new Test();  
  
Test t1=new Test();  
  
System.out.println(t.equals(t1));  
//System.out.println(t==t1);
```

HashCode(): For Every Object, jvm will create a unique no. is called as hascode.

String Operations in Java

What is String in Java?

- >String is group of characters represented in double quotes.
- >String class is Available in java.lang Package.

Ex:

```
String s ="BTPS";
```

Java Provides us Four Imp Classes for String Manipulation

- >java.lang.String : Final class
- >java.lang.StringBuffer : Final Class
- >java.lang.StringBuilder

>java.util.StringTokenizer

>Difference Between String and String Buffer

Case1>String class object can be created by new keyword as well as without new keyword also.

But StringBuffer class object can be created only using new keyword.

Ex:

```
String s=new String("btps"); valid...
```

```
String s1="btps"; valid....
```

Note:

>If you are creating String object using new keyword it will create object in Heap memory.

>can contain duplicate objects

Ex: String s=new String("btps");

```
String s1=new String("india");
```

String s2=new String("btps"); //total 3 objects are created but two are duplicate

>if you are creating String object without new keyword then it will create object in SCP(string constant pool)memory

>can not contain duplicate objects

Ex: String s="btps";

s

Case2>Both the classes are final present in java.lang package.

Case3:String class objects are immutable objects means they are not allowing modification on their content,if we are trying to perform any changes over data then data is allowed for modifications but the resultant data will be stored by creating new object

Where as StringBuffer classs objects are mutable objects,these objects are allowing modification directly on their content.

```
Ex:class Test
{
    Public static void main()
    {
        //immutable object
        String s="btps";
        s.concate("india");
        sop(s); //object data can not modify

        //now creating object of string buffer
        StringBuffer sb=new StringBuffer("btps");
```

```
Sb.append("india")
Sop(sb); //o/p is btpsinida data is modified
```

```
}
```

```
}
```

String Class Constructors

1- String s=new String();

Creates an Empty String object with zero Length.

Ex:- String s="" //creates an empty string object with zero Length like above constrtor

Ex.String s=null //no obect is there in SCP

2- String s=new String(String literal);

Creates a string object on the heap for the given string litral..

3- String s=new String(StringBuffer sb)

Creates an equevalent String object for the given string buffer.

```
Ex:- StringBuffer sb=new StringBuffer("Ramm");
String s=new String(sb);
System.out.println(s);
```

4-String s=new String(Char[] ch)

Creates an equivalent String object for the given character array.

```
Ex:-char [] c={'b','t','p','s'};
String s=new String (c);
Sop(s) //ans. Is... btps
```

5-String s=new String(byte[] b)

Creates an equivalent String object for the given byte array.

```
Ex. byte[] b={97,98,99};
String s=new String(b);
Sop(b); //abc Unicode ....
```

Imp Methods of String Class

1-public char CharAt(int Index);

It returns char locating the specified index

Ex: String s="btps";

Sop(s.charAt(2)); //p

2-public String concat(String s)

```
Ex: String s="btps";
String S1=s.concat("india");
sop(s1); //btpsindia
```

3-public Boolean equals(Object o)

This method is used to perform content comparison where case is important.

This is overridden method of Object class equals method.

4-public boolean equalsIgnoreCase(String s)

This method is used to perform content comparison where case is not important.

Ex:-String s="Btps";

sop(s.equals("BTPS")) //false // to validate pwd of user where case is Imp.

sop(s.equalsIgnoreCase("BTPS")) //true.. To validate user name where case is not imp.

5-public String substring(int beginindex)

It retuns substring from begin index to end of the string.

```
Ex: String s="btpsindia";
sop(s.substring(2)); //psindia
```

6-public String substring(int beginindex,int end index)

It returns substring from begin to end-1 index.

Ex.

**Ex: String s="btpsindia";
sop(s.substring(2,8)); //psindi**

7-public int length()

It returns no. of character present in string.

**Ex. String s="btps";
Sop(s.length()); //4**

If By Mistake we write..

**.String s="btps";
Sop(s.length); //CE:can not find symbol variable length in
java.lang.String**

Note:

**Length variable applicable for array not for string objects whereas
length() is method present in String class.**

8-public String replace(char oldch,char newch)

**Ex.String s="btpsindia";
sop(s.replace(i,z)) //btpszndza**

9-public String toLowerCase()

10- public String toUpperCase()

11- public String trim()

To remove the blankspaces present at begining and end of the string but not middle blank spaces.

12-public int indexOf(char ch)

Returns index of first occurance of specified character.

If specified character is not found in that string then it will return -1

13-public int lastIndexOf(char ch)

Returns index of last occurance of specified character.

Ex.

```
String s="btpsindiaaaa";
Sop(s.indexOf(a)); // 8
Sop(s.lastIndexOf(a)); // 11
```


