Name   : Nitin M
email id   : nitin.m@pw.live
linked id : https://www.linkedin.com/in/nitin-manjunath-110169136/

String
++++++
11.public String trim()
To remove the blank spaces present at the begining and end of string but not the
blank spaces present at the middle of the String.

12.public int indexOf(char ch)
It returns the index of 1st occurance of the specified character if the specified
character is not available then it returns -1.

```
String s="sachinramesh";
System.out.print(s.indexOf('a'));//1
System.out.print(s.indexOf('z'));//-1
```

13.public int lastIndexOf(char ch)
It returns the index of last occurance of the specified character if the specified
character is not available then it returns -1.

```
String s="sachinramesh";
System.out.print(s.lastIndexOf('a'));//7
System.out.print(s.lastIndexOf('z'));//-1
```

=> Because of runtime operation, if there is a change in the content with those changes a new Object will be created only on the heap, but not in SCP.
=> If there is no change then the same object will be reused.
=> This rule is applicable for Objects present in both SCP and Heap.

Note:

```
String name = "";
System.out.println(name.isEmpty());//true
System.out.println(name.isBlank());//true

System.out.println("*****************");

String data = " ";
System.out.println(data.isEmpty());//false
System.out.println(data.isBlank());//true
```

Q>

```
String str =" ";
     str.trim();
     System.out.println(str.equals("")+  " " + str.isEmpty());
```

What is the result?
A. true false
B. true true
C. false true
D. false false

Answer: D(Because String is immutable)

Q> String s = "SACHIN TENDULKAR";
       int len= s.trim().length();
       System.out.println(len);

What is the result?
A. 10
B. 9
C. 8
D. compilation fails
E. 15
F. 16

Answer: F

Q> String s= "Hello world";
       s.trim();
       int i = s.indexOf(" ");
       System.out.println(i);

What is the result?
A. Exception at runtime
B. -1
C. 5
D. 0

Answer: C

Q> String s1= "Java";
   String s2=new String("java");

   //line-1
   {
      System.out.println("equal");
   }
   else
   {
      System.out.println("not equal");
   }
To print equal which code fragment should be inserted?
      A. s1=s2;
          if(s1==s2)
      B. if(s1.equalsIgnoreCase(s2))
      C. String s3= s2;
           if(s3.equalsIgnoreCase(s3))
      D. if(s1.toLowerCase() == s2.toLowerCase())

Answer : A,B,C

9.
 String str = "   PW\tSkills\tPrivateLmtd\tKnown\tfor\tjava   ".strip();
 System.out.println(str);//PW Skills   Private   Lmtd Known for   java

10.
 if("string".toUpperCase() == "STRING"){
      System.out.println(true);

```
 }else{
       System.out.println(false);//false
 }

12.
String str1 = "1";
String str2 = "22";
String str3 = "333";
System.out.println(str1.concat(str2).concat(str3).repeat(3));// 122333122333122333


Mutable
+++++++++
 => If we try to make a change with that change no new object will be created,
changes will happen on the same object.
 => To create a mutable String in java we have 2 classes
      a. StringBuffer
      b. StringBuilder

StringBuffer
===========
   1. If the content will change frequently then it is not recomonded to go for
String object becoz for every new
        change a new Object will be created.
   2. To handle this type of requirement, we have StringBuffer/StringBuilder
concept

1.StringBuffer sb=new StringBuffer();
          creates a empty StringBuffer object with default intital capacity of
"16".
          Once StringBuffer reaches its maximum capacity a new StringBuffer Object
will be created
                new capacity = (currentcapacity+1) * 2;

eg1::StringBuffer sb = new StringBuffer();
     System.out.println(sb.capacity());//16
     sb.append("abcdefghijklmnop");
     System.out.println(sb.capacity());//16
     sb.append('q');
     System.out.println(sb.capacity());//34

2. StringBuffer sb=new StringBuffer(initalCapacity);
    It creates an Empty String with the specified inital capacity.
eg1::StringBuffer sb = new StringBuffer(19);
     System.out.println(sb.capacity());//19

3. StringBuffer sb=new StringBuffer(String s);
      It creates a StringBuffer object for the given String with the capacity =
s.length() + 16;
eg1::StringBuffer sb = new StringBuffer("sachin");
     System.out.println(sb.capacity());//22
```