for INTEGER buffer is -127to 127
For other data types also same Range sir?
      For all Wrapper classes buffer range is of byte only.
      Character -> 0 to 127
        Boolean   -> true, false

sir, diff. b/w buffer and array r they same?
      Array -> objects which we should create
      buffer-> programmer won't create rather jvm will create for performance.

sir u said that wrapper classes are immutable but how value of x is changed in code snippet that u explained
     Integer x= 10;(immutable)
     Integer y =x;(immutable)
         x++;//x = x+1(since it is immutable change won't happen in the same object
                          rather change will happen and new object will be created)

for boolean type, in a java class when we write boolean/Boolean, both are considered same,
which is not the same with other datatypes, why?
      That is the convention Oracle team /SunMicroSystem team had followed.

one interview qst i got..
wap to get only integer number if any other number or char it should throw error...
in short only integer is allowed
            logical question through if else.

in string you have SCP like we have for Wrapper classes also
    String => SCP
   Wrapper class -> Buffer of range byte

sir can u explain about capacity method in StringBuffer class
    StringBuffer sb=new StringBuffer();
      System.out.println(sb.capacity());//16(it indicates the no of characters which can be stored)

  StringBuffer sb=new StringBuffer("sachin");
      System.out.println(sb.capacity());// capacity = 16 + length of String
                                                                  16 + 6
                                                                = 22

StringBuffer sb=new StringBuffer(10);
    System.out.println(sb.capacity());//capacity = 10

If we use value of method explicitly buffer won't be created right?
    Integer i1 =new Integer(10);
    Integer i2 =new Integer(10);
      System.out.println(i1==i2);//false
                  vs
   Integer i1=10;//Autoboxing(valueOf())
   Integer i2=10;//Autoboxing(valueOf())
      System.out.println(i1==i2);//true

Sir buffer which jvm is creating and the StringBuffer class which we used in Mutable concept are same or different?
          Buffer -> Wrapper class and StringBuffer Class buffer are different.

what are different ways to create objects?
    a. new
    b. instanceOf()
    c. Using Factory methods like valueOf()
    d. using clone()

Integer i1=10; will this create any object?
   First time it will use the object already available in Buffer.

sir for string we have same value it will refer same object but in wrapper class if we have same value using new keyword is different?

```
    String s1=new String("sachin");
    String s2=new String("sachin");
                  vs
    Integer i1=new Integer(10);
    Integer i2=new Integer(10);
```

Question
sir interface methods implemented in class those methods by default public or default in class..
and interface methods implemented in abstract class those methods are by default public or
default in abstract class becoz in abstract class all the methods are public by default

```
abstract class Bank{'
      // it is default
      abstract void depositAmount();
      abstract long withDrawAmount();
      abstract String giveCheque();

      void giveNotification(){
            //implementation
      }
}
            vs

interface Bank{
      //abstract and public
      void depositAmount();
      long withDrawAmount();
      String giveCheque();
}
```

Will the valueOf() create an object in buffer ?
      it will use the Object, but can't create an object inside buffer

```
Double c=2.0;//not in range of byte
Double cc=2.0;//not in range of byte
System.out.println(c==cc); It's giving false
```

Sir,In Eclipse can we run the program in .class file without .java file?
      yes possible using war/jar file approach using maven tools..

Integer i1=10 by default it is valueof(10) to convert primitive to wrapper class or by default it is
      Integer i1=new Integer(10)?
    Integer i1=10;

```
         compiler will make
    Integer i1=Integer.valueOf(10);
```

is there any way to increase the size of buffer like we have ensureCapacity in strings
   Not possible

String is immutable how can you explain with Memomery also?

```
    String s=new String("sachin");
      s.concat("tendulkar");
    System.out.println(s);// sachin
```

when auto unboxing comes into picture?

```
      Integer i1=new Integer(10);
      int i2=i1;
                  or
public void m1(int i){//AutounBoxing
      System.out.println(i);
}
  Integer i=10;//autoboxing
      m1(i);
```

Question
string is an type of object and wrapper is also used to wrap string or primitive to an object,
then both things are quite similar,can u please explain me

```
String firstInput = "10";//String
String secondInput = "20";//String
String result = Integer.value(firstInput)+ Integer.valueOf(secondInput);
System.out.println(result);//30
```

wrapper buffer is part of stack or heap?
   heap.

How to print address of object?

```
 Integer i = new Integer(10);
     System.out.println(i);//i.toString() prints data
```

```
String s1= new String("sachin");
String s2=s1.concat("tendulkar");
System.out.println(s1);//sachin
System.out.println(s2);//sachintendulkar
```

Question

```
String str1="10";
String str2="20";
String res1 = Integer.valueOf(str1) + Integer.valueOf(str2);//CE
```

Yesterday you come with 2 interface class with same signature implementing class had override function with super class could you explain it once again again sir?. because i missed it some where

```
interface Right{
      default void m1(){
            System.out.println("hiee");
      }
}
interfae Left{
```

```
	default void m1(){
		System.out.println("hello");
	}
}
class TestImpl implements Left,Right{

	@Override
	public void m1(){
		System.out.println("byee");
		Left.super.m1();//hello
		Right.super.m1();//hiee
	}
	public static void main(String[] args){
		Left l =null;
		l.m1();//CE

		TestImpl t = new TestImpl();
		t.m1();//byee

		Left l =new TestImpl();
		l.m1();//byee
	}
}

interface Right{
	default void m1(){
		System.out.println("hiee");
	}
}
interfae Left{
	default void m1(){
		System.out.println("hello");
	}
}
class TestImpl implements Left,Right{}//Compile Time Error


can u explain casting between wrappers..?
     Casting to happen we need to have parent child relationship, but wrapper
classes are siblings for Number class.

interface Left{
	public void m1();
}
interface Right{
	public void m1();
}
public class TestImpl implments Left,Right{
	@Override
	public void m1(){

	}
}

public final class Integer extends Number
Integer is extending Number class then how they are siblings
       Number
		|-> Byte
		|-> Short
```

```
|-> Integer
```