Sir, can you explain how to create multiple objects for a class, if no of Objects is coming from console?

```
class Test{
      ;;;;
}
class Demo{
      public static void main(String[] args){
            Scanner scan =new Scanner(System.in);
            System.out.println("Enter the no of objects to be created: ")
            int number = scan.nextInt();
            for(int i = 0;i<number;i++)
                  Test t1 = new Test();
      }
}
```

In normal inner classes Static method is possible when i say java Outer$Inner i getting the output how? i sented the code to you in dicord also.

      Answer: higher version u can use static also.

```
class Outer{
      class Inner{
                  public static void main(String[] args)
                  {
                        System.out.println("Inner class main method");
                  }
      }
      public static void main(String[] args)
      {
            System.out.println("Outer class main method");
      }
}
```

```
javac Outer.java
java Outer
   Outer class main method
java Outer$Inner
   Inner class main method
```

Enum class also extends Object class?

  Answer : yes

Q> can we call static nested class main method from outer class main method by any ways..?

```
class Outer
{
      static class Nested
      {
                  public static void main(String[] args)
                  {
                        System.out.printnl("Hello");
                  }
      }
      public static void main(String[] args)
      {
                  System.out.println("hiee");
                  Nested.main(new String[]{""});
      }
}
```

```
javac Outer.java
java Outer
```

sir if you recall the method local inner class, can you again discuss regarding the type of variables we will
be able to access?

```
class Test
{
       int i=10;
       static int j=20;
        public void m1()
        {
             int k=30;
             final int m=40;
               class Inner
                 {
                    public void m2()
                     {
                            LINE 1;
                      }
                 }
      }
}
m1 instance -> i,j,k,m
m1 is static ->  j,k,m
m2 is static -> CE
```

```
import java.util.*;
class TestGenerics1{
      public static void main(String args[]){

            ArrayList name=new ArrayList();
            name.add("ASD");
            name.add("A");
            name.add("FG");
            name.add(new Integer(12));
            name.add(new Float(12.5));

            System.out.println(name.get(0));
            System.out.println(name.get(1));
            System.out.println(name.get(2));
            System.out.println(name.get(3));
            System.out.println(name.get(4));

}
```

sir in last class you told that an abstract class can have inner class, but how can we access
inner class because it cannot be access without creating an object of outer class.

```
//public,default,strcitfp,final,abstract
abstract class Outer{
      class Inner
      {

      }
}
//give the implementation for all the abstract methods and then we need to access.
```

what's mostly used abstract class or interface ?? I've watched recorded videos multiple times but still getting silly doubts about abstraction vs interface
Answer:

```
interface
     |
     |
     |
abstract class
     |
     |
     |
  class
```

```
class Launch2 {
          int age = 19;
       public void display() {
           int age = 18;
           System.out.println(age);
            class Sub {
                  int age = 17;
                  public void get() {
                        int age = 16;
                        System.out.println();
                        System.out.println("get method:: " + age);
                        System.out.println("Method local inner class:: " +
this.age);
                        System.out.println("Outer class:: " + Launch2.this.age);
                  }
            }
            new Sub().get();
      }
    public static void main(String[] args) {
          new Launch2().display();
    }
}
```

```
int method1(){
      int a=10, b= 20;
      return a+b;
}
```

sir where will this 30 be stored,, in heap or in stack
Answer:stack becoz it is local to method

```
enum ExamResult
{
      PASS(1),FAIL(0),ABSENT(-1),PENDING;

      int result;
      ExamResult(int result)
      {
            System.out.println("Parameterized constructor getting called");
            this.result =result;
      }
      ExamResult()
      {
            this.result = -100;
      }
      public int getResult()
      {
            return result;
      }
}

/*
      final class ExamResult extends java.lang.Enum
      {
            public static final ExamResult PASS=new ExamResult(1);
            public static final ExamResult FAIl=new ExamResult(0);
            public static final ExamResult ABSENT=new ExamResult(-1);
            public static final ExamResult PENDING=new ExamResult();

            int result;
            ExamResult(int result)
            {
                  System.out.println("Parameterized constructor getting called");
                  this.result =result;
            }
            ExamResult()
            {
                  this.result = -100;
            }
            public int getResult()
            {
                  return result;
            }
      }
*/
class TestApp {
      public  static void main(String[] args) {
            ExamResult[] results =ExamResult.values();

            for (ExamResult result: results )
            {
                  System.out.println(result+":::"+result.getResult());
            }
      }
}
```

Sir I have written the below logic to find duplicates in array, but if a number is present
more than twice, the message is printed more than once, can you please let me know the logical mistake I'm makking?

```
int[] a ={10,20,30,20,40};
for (int i = 0; i < len; i++) {
      for (int j = i + 1; j < len; j++) {
                  if (a[i] == a[j])
                  {
                              count++;
                  }
            if(count>0){
                  System.out.println(a[i]+" is present in the provided array time
at multiple times");
                  count = 0;
            }
      }
}
```

Q>

```
int[]a = {0,2,4,1,3};
for(int i=0; i<a.length; i++){
      a[i] = a[(a[i]+3)/a.length];
}
s.o.p(a[1]);//2

a[0] = 0
a[1] = 2
a[2] = 4
a[3] = 1
a[4] = 3

a[0] = a[3/5]
a[0] = a[0]

a[1] = a[5/5]
a[1] = a[1]
```