Q>
can you please tell what this is error cannot find symbol?

```
String name = "sachin";
System.out.println(name.length);//CE: can't find symbol :: length
```

Q>
In Enum, Why was the constructor called 3 times for-(Fail,Pass, NoResult), when we were only using PASS as reference ?

```
enum EResult{

      // public static final EResult PASS =new EResult();
      // public static final EResult FAIL =new EResult();
      PASS,FAIL;
      EResult(){
            System.out.println("constructor of enum is called");
      }
}
public class Test {
      public static void main(String[] args) {

            EResult eResult =null;
            eResult=EResult.PASS;

      }
}
```

Sir, I have a doubt. When we are creating an object, in the heap it will be created and its instance variables also in heap. But what about the methods present in the object sir? Will it be created only once it is called and those methods will find space in stack? Is this correct sir?

those named constants are refernce of enum type, so only object is created.

Q>
Sir, I have a doubt. When we are creating an object, in the heap it will be created and its instance variables also in heap.
But what about the methods present in the object sir?
Will it be created only once it is called and those methods will find space in stack? Is this correct sir?

method area -> .class file and static varaible memory(jvm will give default value as per the datatype)
stack area   -> local variables will get memory followed by the code which needs to executed by JVM
heap area   -> Object data(instance variable) will get memory and jvm will give default value as per the datatype

Q>
@FuncationalInterface annotation used in interface its allowed to only one abstract method ,
if more than one abstract method try to add its shows error, like that I had requirement to Interface to allow
only two abstract method , is it possible to create any custom annotation to achieve this?

Answer: yes possible with custom annotations.(too complex)

Q>
Sir I was trying to input values with for each:Source codeimport java.util.*;

```
public class VarargsDemo {
      public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            int n=5;
            int nums[]=new int[n];

            //foreach loop should not be used for writing the data to array.
            for(int i:nums) {
                  System.out.println("Enter number:");
                  i=sc.nextInt();
            }
            //foreach loop is used to read the data from array
            for(int j:nums) {
                  System.out.println(j+"\tnumber:"+nums);
            }
            Calc obj=new Calc();
            int result=obj.add(nums);
            System.out.println(result);

      }

}

class Calc{
      int sum=0;
      public int add(int... nums) {
            for(int i:nums) {
                  sum=sum+i;
            }
            return sum;
      }
}Output is:Enter number:
11
Enter number:
12
Enter number:
23
Enter number:
34
Enter number:
23
0     number:[I@62ee68d8
0     number:[I@62ee68d8
0     number:[I@62ee68d8
0     number:[I@62ee68d8
0     number:[I@62ee68d8
0What is wrong?
```

Q>
Sir one of the advantage of collection (ArrayList over) Arrays is it store heterogeneous data but when Generic
was introduced in JDK 1.5 then this advantage is no longer but still we have one advantage of it is it is
dynamic so we can add element to it whereas in Arrays it is fixed size that is if we want to add element to it then we have to create a new Array object
Sir the statement I had wrote is correct?

Answer: Correct

Q>
how to use cmd to extract info about pakages and interface? if i try i got error

```
    javap fullyqualified classname

      eg: javap java.lang.Object
```

Q>
what is fill ratio??

fillration-> upon how much filling the data into the collection, i need to increase my capacity is specified through fillratio.

Q>
Can you explain :: (double colon) operator and use?

:: -> method reference and also has constructor reference.
usage method reference can be replace in lambda expression also

```
                  eg: ArrayList<integer> al =new ArrayList<Integer>();
                                    al.add(0); al.add(10); al.add(5);
                        al.stream().forEach(System.out::print);
```

Innerclass --> required in SpringBoot projects
filehandling---> without knoweldge of this io can't be attained(JDBC)
Serialization,DeSerialization -> Hibernate and JDBC internaly work can't be understood.