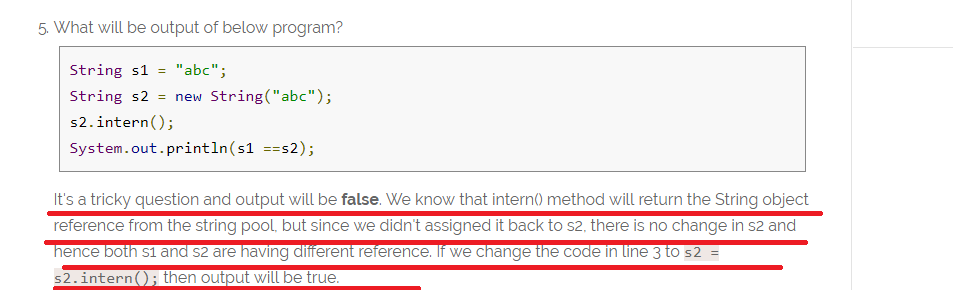
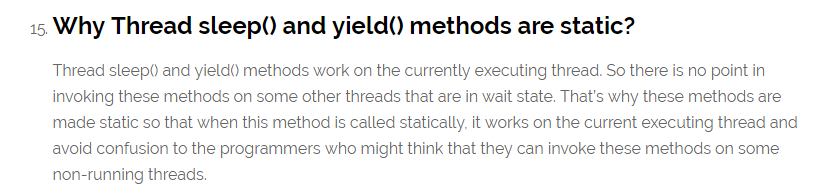
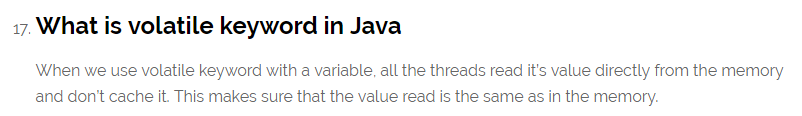
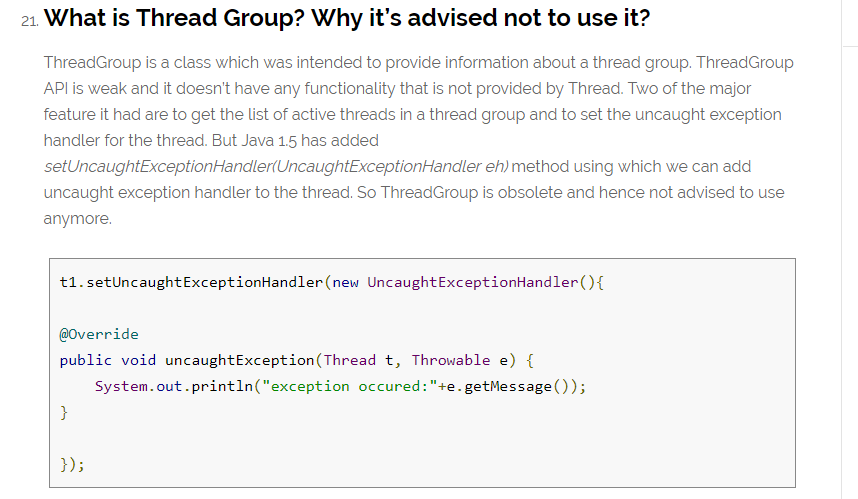
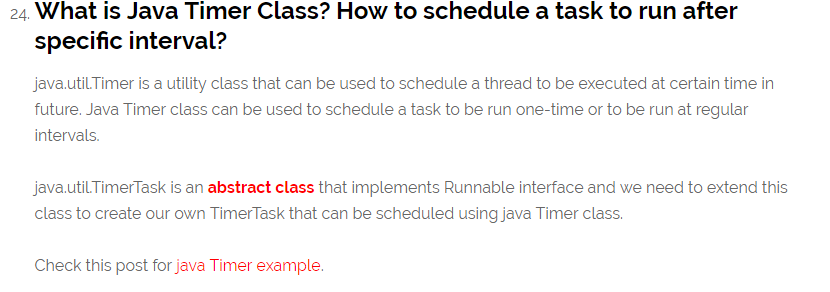
Journal Dev- String

1. 
2. System.out.println("substring equals subSequence ? " + (str.substring(4, 14).equals(str.subSequence(4, 14))));

For the point 2 output is “**true”**

Journal Dev- Thread

Part -1 🡺 Basic

1. 
2. 
3. Java ThreadLocal is used to create thread-local variables.
4. 
5. Avoid Nested Locks, Lock Only What is Required and Avoid waiting indefinitely are common ways to avoid deadlock situation, read this post to learn how to [analyze deadlock in java](https://www.journaldev.com/1058/deadlock-in-java-example) with sample program. 🡺 **needed practical example on how deadlock created and how it can be avoid**
6. 

**java.util.TimerTask** is an [**abstract class**](https://www.journaldev.com/1582/abstract-class-in-java) that implements Runnable interface and we need to extend this class to create our own **TimerTask** that can be scheduled using java Timer class.

package com.journaldev.threads;

import java.util.Date;

import java.util.Timer;

import java.util.TimerTask;

public class MyTimerTask extends TimerTask {

@Override

public void run() {

System.out.println("Timer task started at:"+new Date());

completeTask();

System.out.println("Timer task finished at:"+new Date());

}

private void completeTask() {

try {

//assuming it takes 20 secs to complete the task

Thread.sleep(20000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

public static void main(String args[]){

TimerTask timerTask = new MyTimerTask();

//running timer task as daemon thread

Timer timer = new Timer(true);

timer.scheduleAtFixedRate(timerTask, 0, 10\*1000);

System.out.println("TimerTask started");

//cancel after sometime

try {

Thread.sleep(120000);

} catch (InterruptedException e) {

e.printStackTrace();

}

timer.cancel();

System.out.println("TimerTask cancelled");

try {

Thread.sleep(30000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

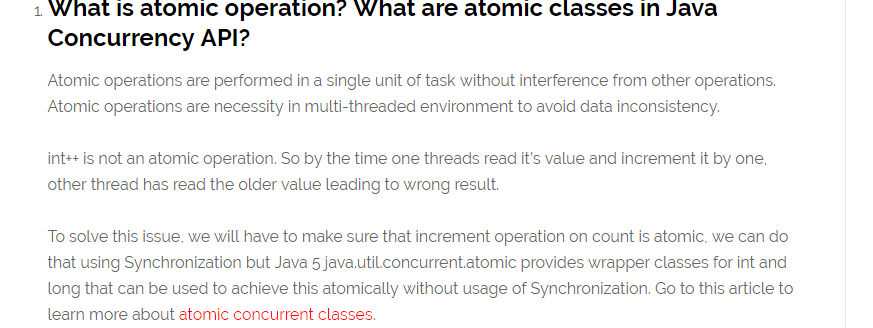
Notice that one thread execution will take 20 seconds but Java Timer object is scheduled to run the task every 10 seconds. Here is the output of the program:

Timer cancel() method is used to terminate the timer and discard any scheduled tasks, however it doesn’t interfere with the currently executing task and let it finish. If the timer is run as [daemon thread](https://www.journaldev.com/1072/daemon-thread-in-java), whether we cancel it or not, it will terminate as soon as all the user threads are finished executing.

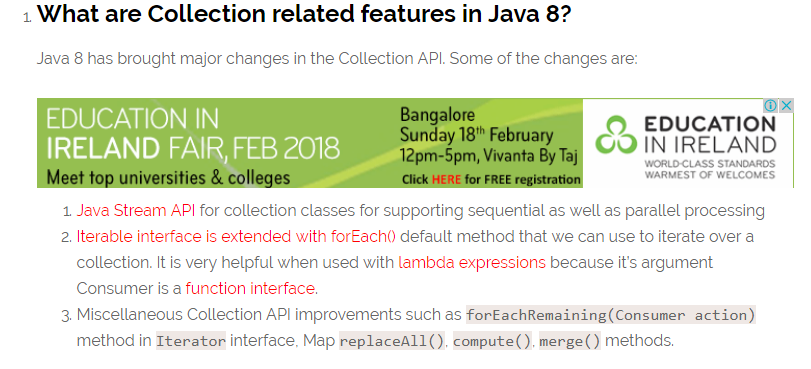
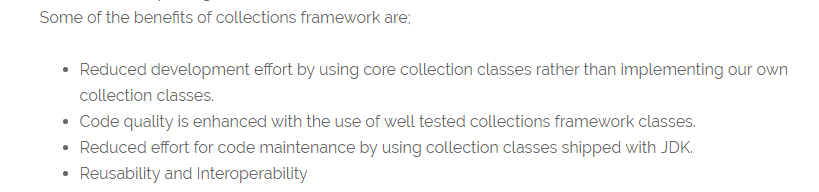
Timer class contains several **schedule**() methods to schedule a task to run once at given date or after some delay. There are several **scheduleAtFixedRate**() methods to run a task periodically with certain interval

While scheduling tasks using Timer, you should make sure that time interval is more than normal thread execution, otherwise tasks queue size will keep growing and eventually task will be executing always. That’s all for a quick roundup on Java Timer and Java TimerTask.

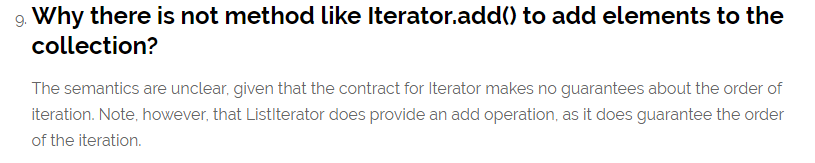
## Part 2: Java Concurrency Interview Questions and Answers

1. 
2. **NOTE FEW MORE QUESTIONS ARE PRESENT ON CONCURRENCY BUT ITS QUITE HIGH LEVEL**

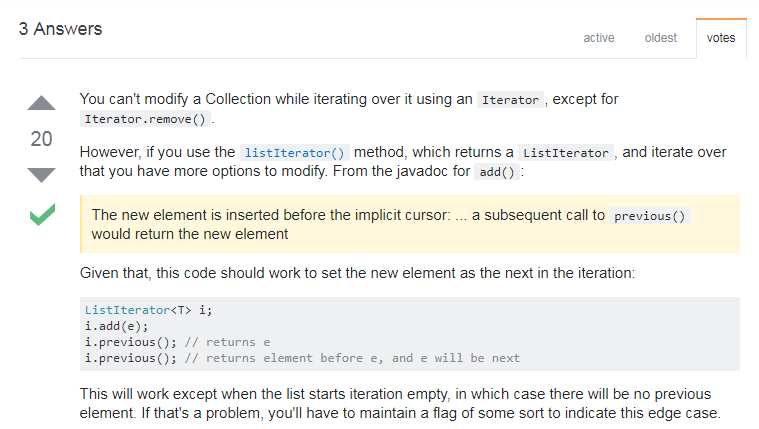
JOURNALdEV-Collection

1. 
2. 

### What is the purpose of  Cloneable and Serializable interfaces?

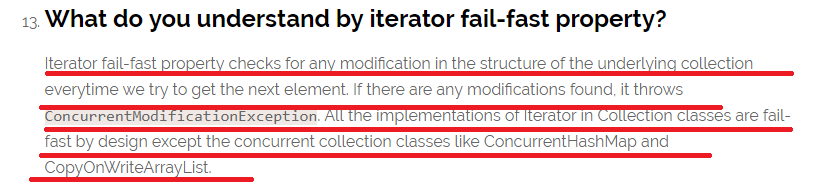
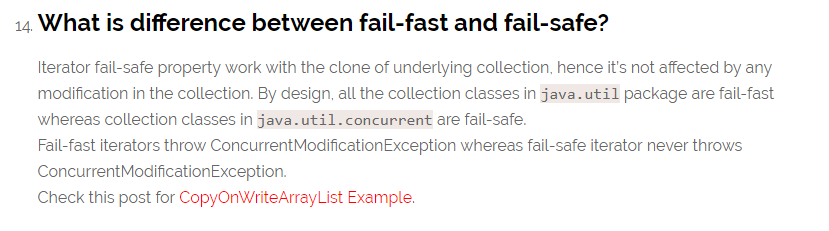
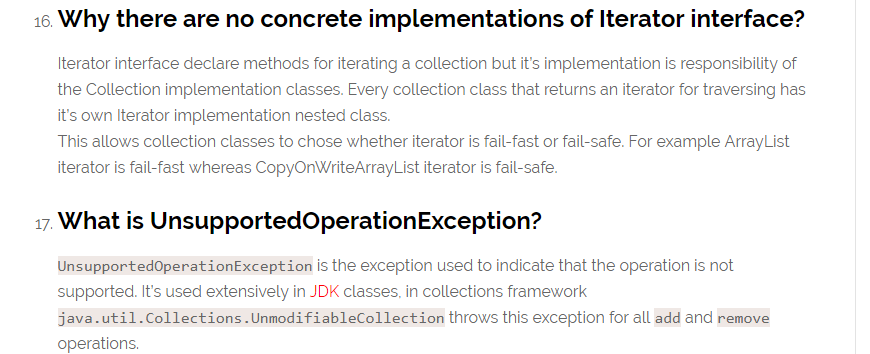
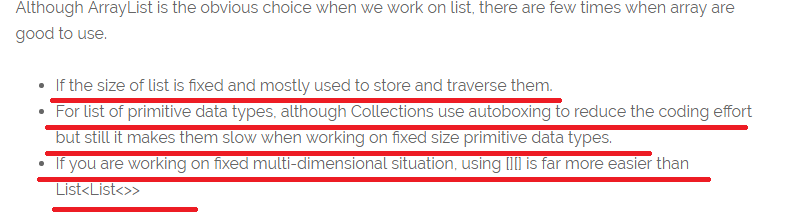
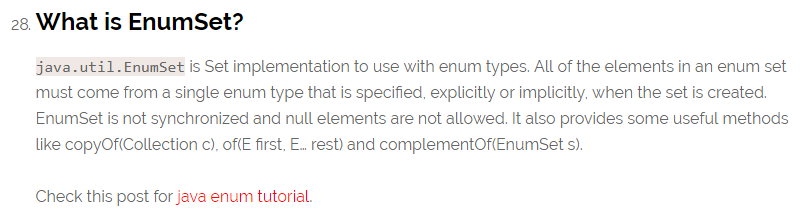
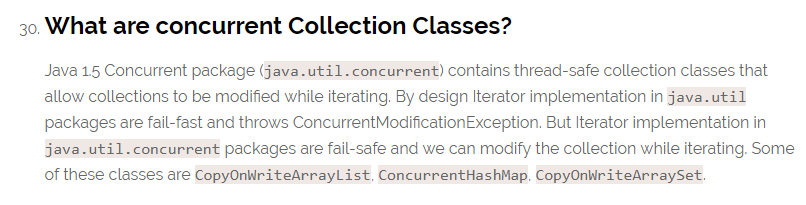
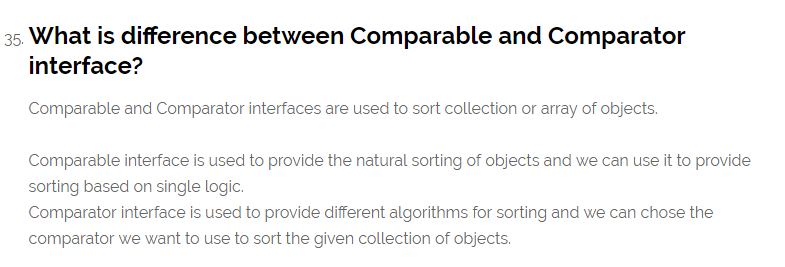
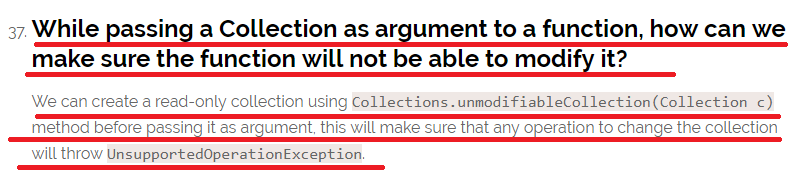
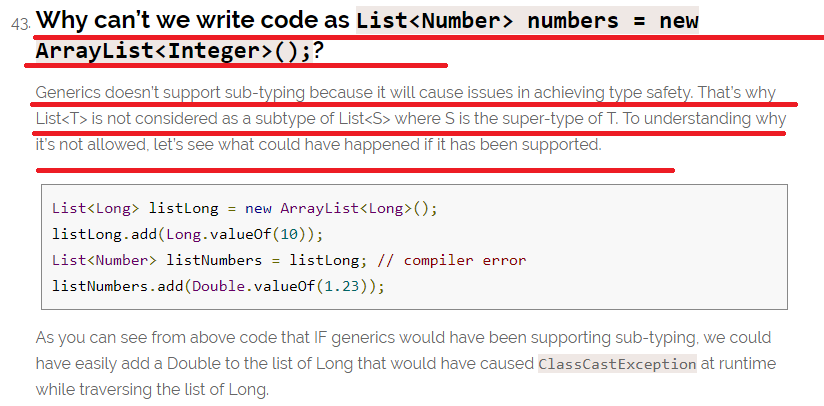
1. 

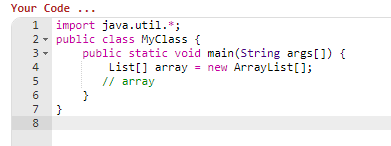
Note 🡺 from Stack Overflow

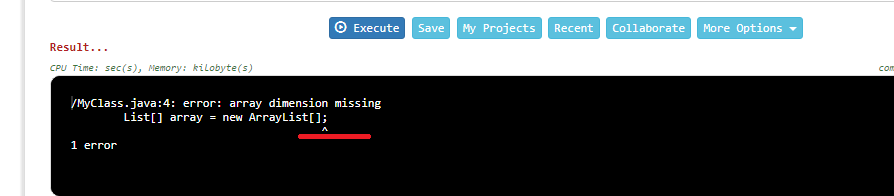


### What are different ways to iterate over a list?

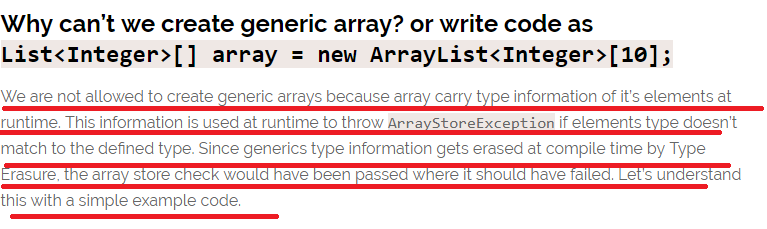
We can iterate over a list in two different ways – using iterator and using for-each loop.

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. What is the Output of the following code?



Output 🡺

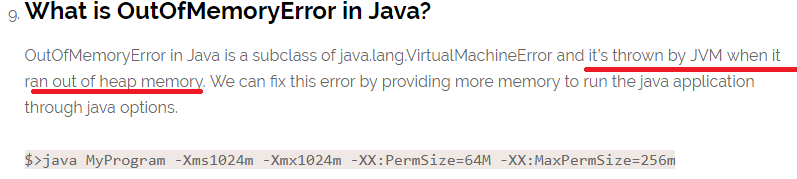
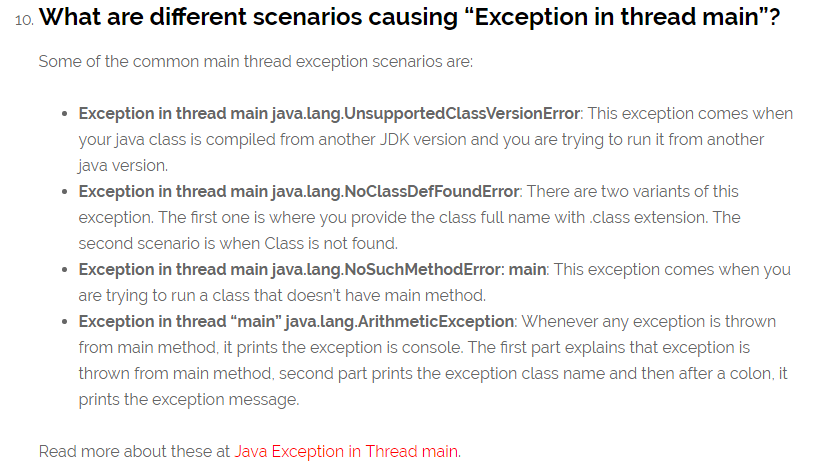
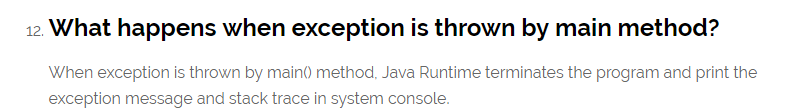
**See here though array created is of Collection type it requires fixed Size.**

1. 

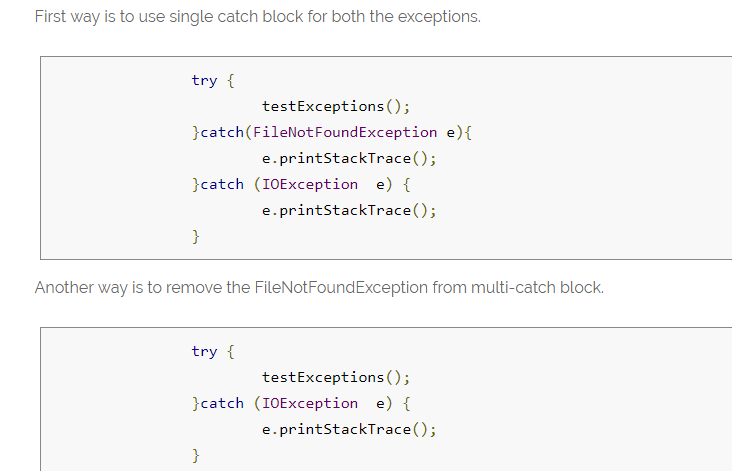
Journal Dev- Exception Interview Questions

1. Exception is an error event that can happen during the execution of a program and disrupts it’s normal flow.

### Explain Java 7 ARM Feature and multi-catch block? 🡺

1. 
2. What about StackOverFlow?
3. 
4. final keyword can be used with class variables so that they can’t be reassigned
5. 
6. 

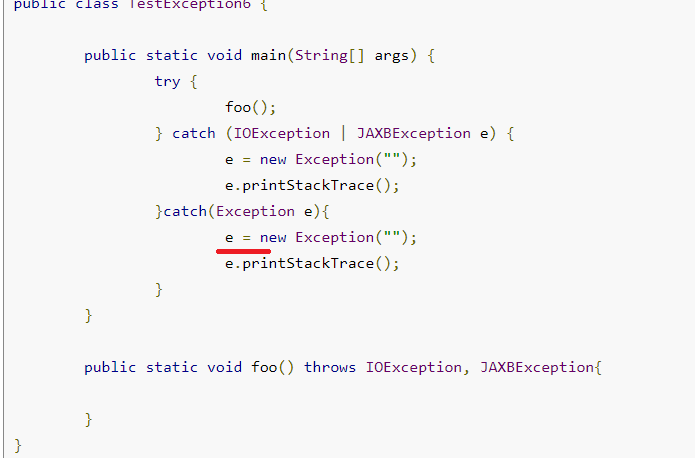
Above program won’t compile and you will get error message as “The exception FileNotFoundException is already caught by the alternative IOException”. This is because FileNotFoundException is subclass of IOException, there are two ways to solve this problem.



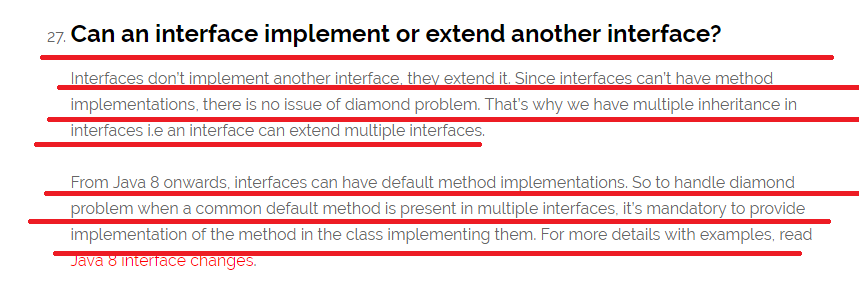
1. What is the output for the below program?



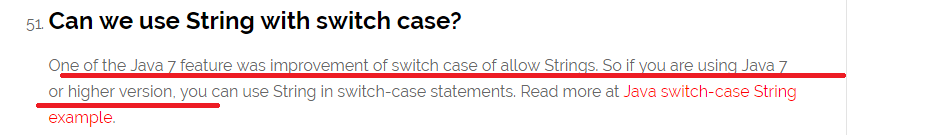
**The program won’t compile because JAXBException is a checked exception and foo() method should throw this exception to catch in the calling method. You will get error message as “Unreachable catch block for JAXBException. This exception is never thrown from the try statement body”. 🡺 To solve this issue, you will have to remove the catch block of JAXBException.**

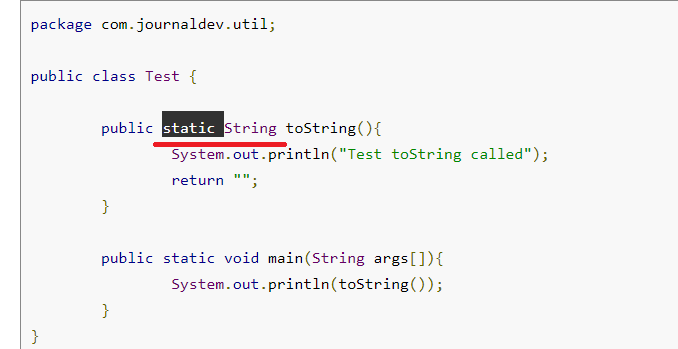
1. **exception object in multi-catch block is final and we can’t change it’s value.  🡺 is from Java 7**
2. What is the output for the below code? 🡺the red colour underlined line throws an exception

Now Let’s see the Journal Dev general Code Java Questions

1. 
2. Java Reflection API provides ability to inspect and modify the runtime behavior of java application

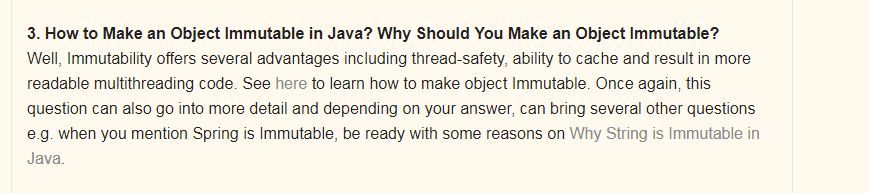
### How to sort a collection of custom Objects in Java? 🡺 without using the build function

1. 
2. What is the output for the following code🡺

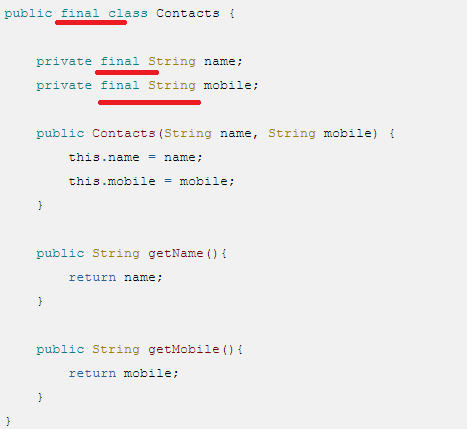


**Answer: The code won’t compile because we can’t have an Object class method with static keyword. Note that Object class has toString() method. You will get compile time error as “This static method cannot hide the instance method from Object”. The reason is that static method belongs to class and since every class base is Object, we can’t have same method in instance as well as in class. You won’t get this error if you change the method name from toString() to something else that is not present in super class Object.**

Interview Questions from 🡺 <http://www.java67.com/2013/07/15-advanced-core-java-interview-questions-answers-senior-experienced-5-6-years-programmers-developers.html>

1. 

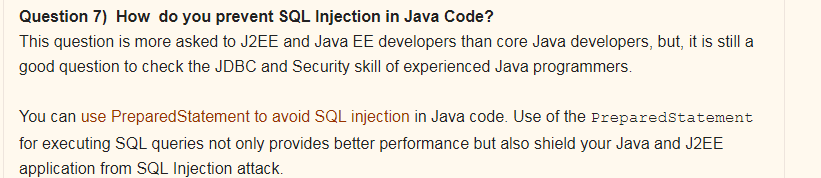
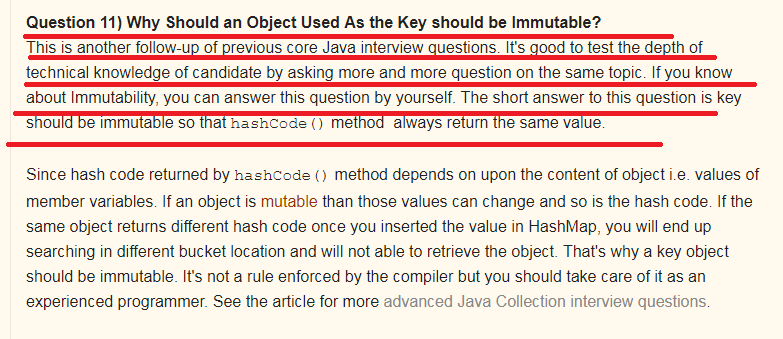
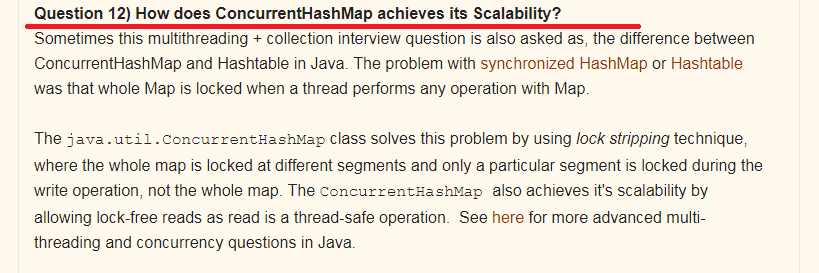
NOW LETS SEE HOW TO MAKE AN OBJECT IMMUTABLE🡺 WHAT IS THE DIFFERENCE BETWEEN MAKING A CLASS AS IMMUTABLE OR MAKING AN OBJECT AS IMMUTABLE

1. Syntax to create a immutable class

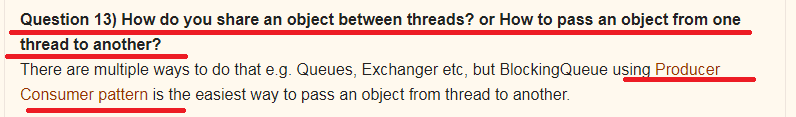
Explanation 🡺

1. This Java class is immutable, because its state can not be changed once created.
2. You can see that all of its fields are final. 🡺 NOTE WE HAD MARKED IN THE CODE AS FINAL BUT I DID HANDS ON THAT IT WAS NOT FINAL BY DEFAULT,   
     
   Read more: <http://javarevisited.blogspot.com/2013/03/how-to-create-immutable-class-object-java-example-tutorial.html#ixzz53s9qcmAf>

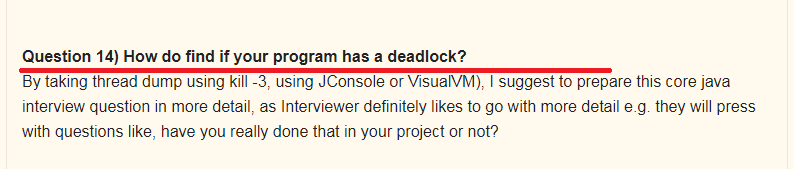
Doubt 🡺

1. Which Design Patterns have You Used in Your Java Project?  
     
   Always expect some design patterns related question for Core Java Interview of senior developer position. It's a better strategy to mention any GOF design pattern rather than Singleton or MVC, which almost every other Java developer use it.   
     
   Your best bet can be [Decorator pattern](http://java67.blogspot.sg/2013/07/decorator-design-pattern-in-java-real-life-example-tutorial.html) or may be [**Dependency Injection Pattern**](http://javarevisited.blogspot.sg/2012/12/inversion-of-control-dependency-injection-design-pattern-spring-example-tutorial.html)**, 🡺 NOTE WE CAN TALK ABOUT THE Dependency Injection Design Pattern.**
2. 
3. 
4. 

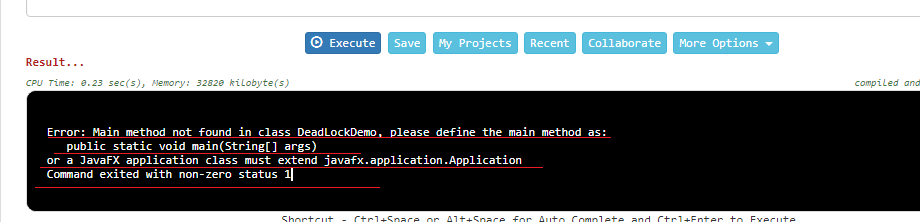
Explanation🡺

1. 

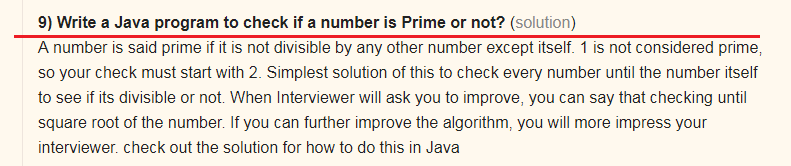
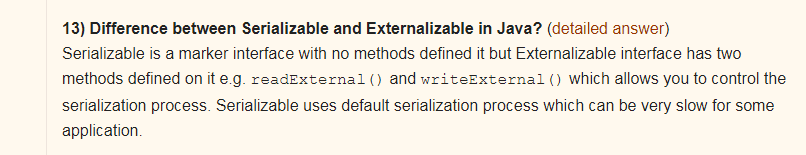
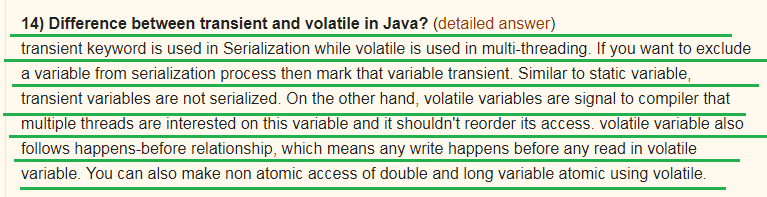
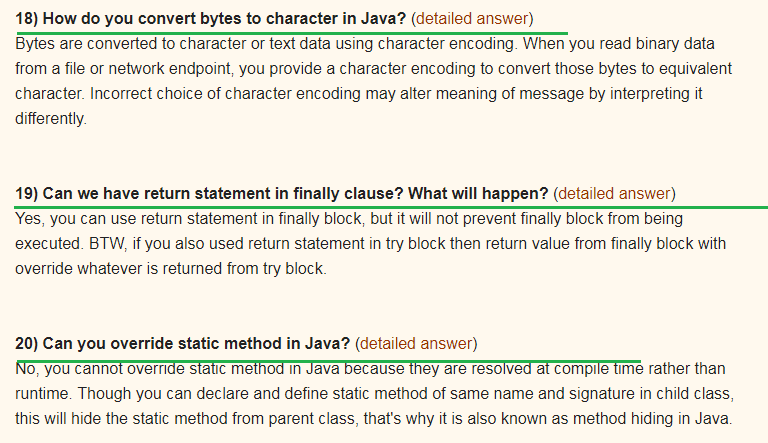
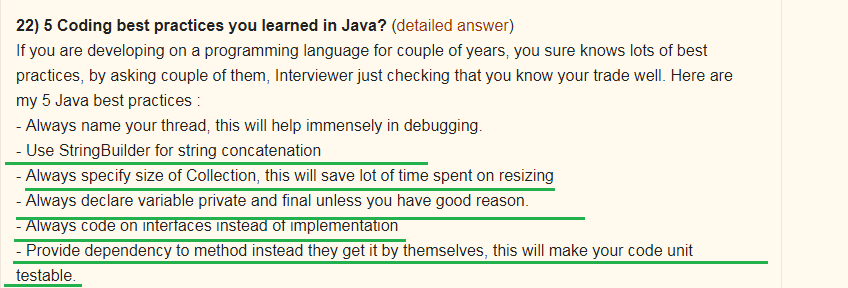
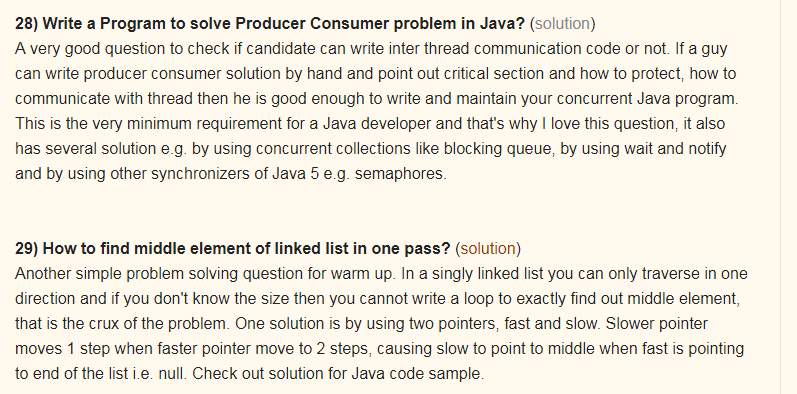
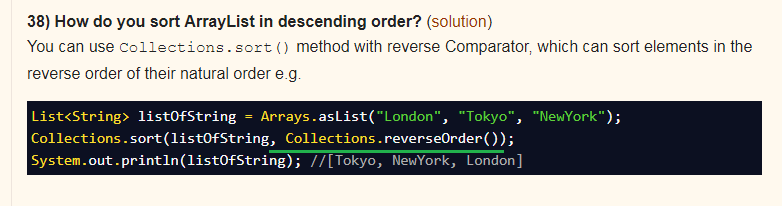
Explanation🡺

1.  Explanation:
2. Write a Java program which will result in deadlock?  
   

**See here usually we have seen the syntax of methodName(Datatype variable) but here it is in the format methodName(ClassNAme.class)**

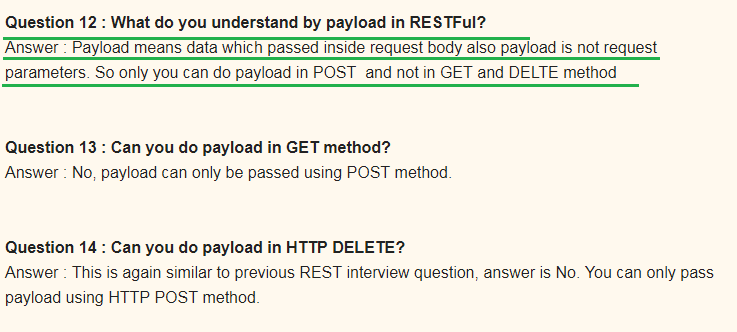
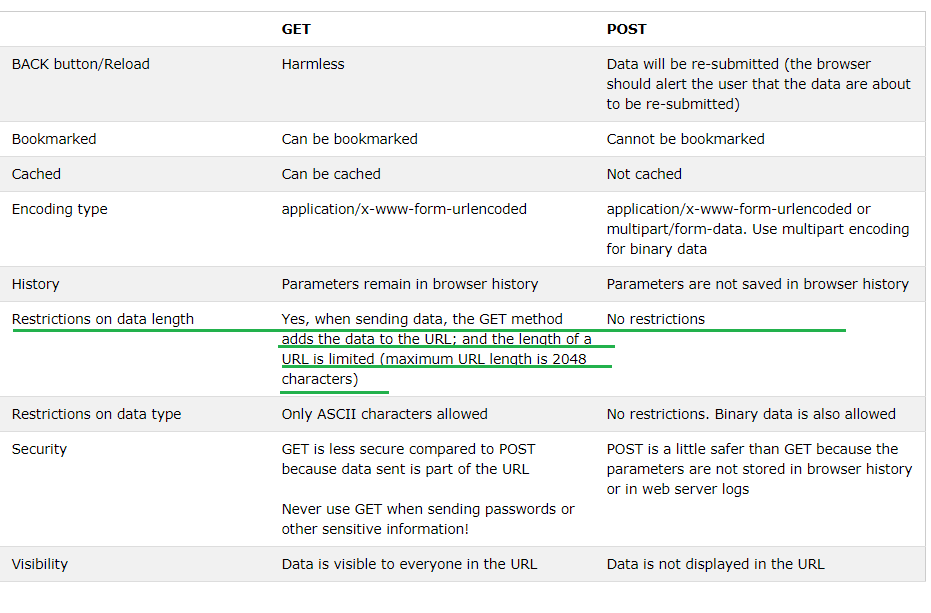
1. What will be the Output if we run the Application Without Main Method🡺

Core Java Interview Questions 🡺 from 🡺 <http://www.java67.com/2015/03/top-40-core-java-interview-questions-answers-telephonic-round.html>

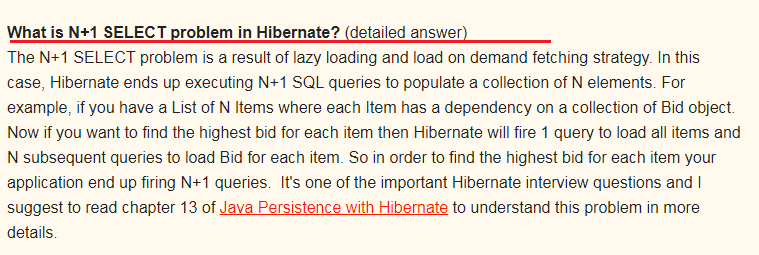
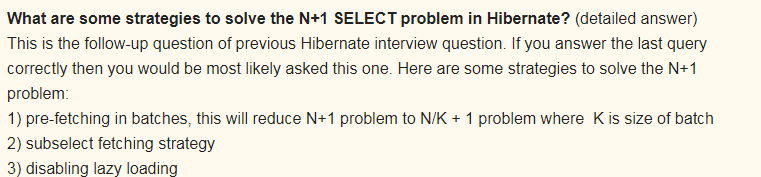
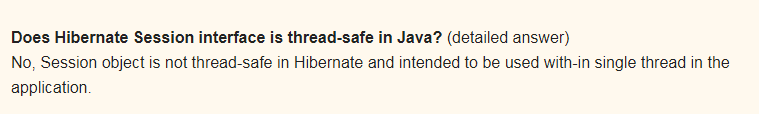
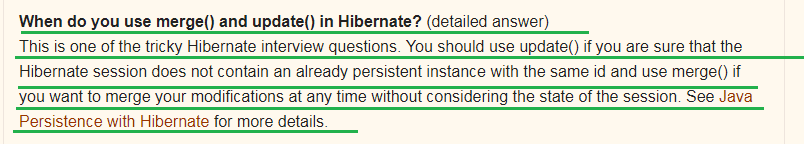
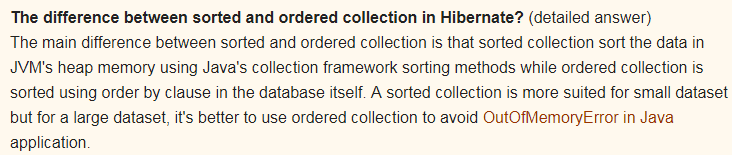
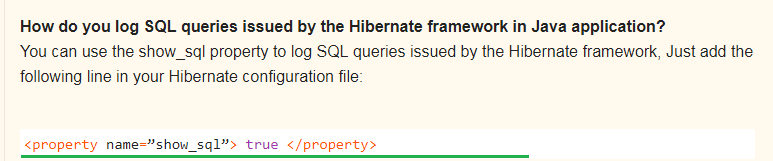
1. Logic for Fibonacci series 🡺 sum of previous two numbers i.e. f(n) = f(n-1) + f(n-2).
2. 
3. **11) How to check if linked list contains loop in Java?** ([solution](http://javarevisited.blogspot.sg/2013/05/find-if-linked-list-contains-loops-cycle-cyclic-circular-check.html))
4. 
5. 
6. 
7. 
8. 
9. **28) Write a Program to solve Producer Consumer problem in Java?** ([solution](http://java67.blogspot.sg/2012/12/producer-consumer-problem-with-wait-and-notify-example.html))
10. **29) How to find middle element of linked list in one pass?** ([solution](http://javarevisited.blogspot.sg/2012/12/how-to-find-middle-element-of-linked-list-one-pass.html))
11. 

Rest Based Interview Question from 🡺 <http://www.java67.com/2015/09/top-10-restful-web-service-interview-questions-answers.html>

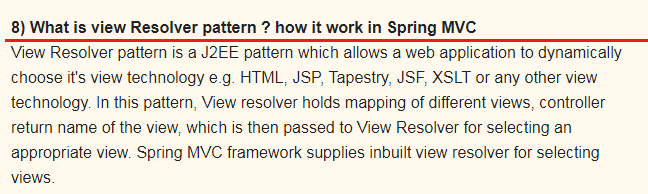
Rest Service Questions

1. Question 3: What is HTTP Basic Authentication and how it works?
2. Question 4: Can you tell me which API can be used to develop RESTFul web service in Java? 🡺 There are many framework and libraries out there which helps to develop RESTful web services in Java including JAX-RS which is standard way to develop REST web services. Jersey is one of the popular implementation of JAX-RS which also offers more than specification recommends. Then you also have RESTEasy, RESTlet and Apache CFX. If you like Scala then you can also use Play framework to develop RESTful web services
3. Question 5 : How do you configure RESTFul web service?
4. Question 8 : How you maintain session in RESTful services?
5. 
6. 

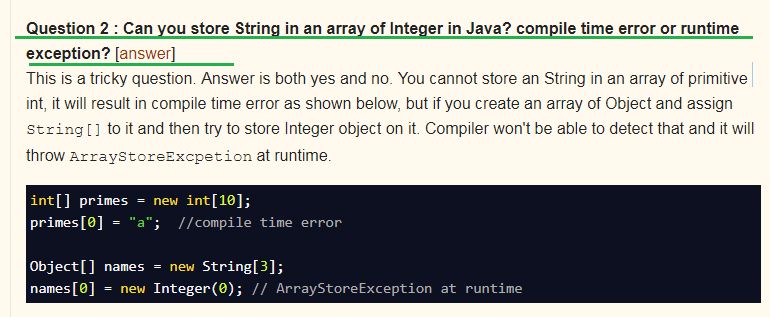
Top 20 Hibernate Interview Questions for Java J2EE Programmers 🡺 <http://www.java67.com/2016/02/top-20-hibernate-interview-questions.html>

1. 
2. 
3. 
4. 
5. 
6. 

Top 23 Spring MVC Framework Interview Questions Answers - Java JEE 🡺 <http://www.java67.com/2012/08/spring-interview-questions-answers.html>

1. 
2. 15) Can we use more than one configuration file for our Spring project?
3. 18) Can you use Spring MVC framework along with Struts ? I have an existing Java MVC application which is based in Struts, Can I migrate that to use Spring MVC ? **How ?**
4. 19) What is the advantage of Spring MVC framework over Struts 1.0 or Struts 2.0 ? is it worth to convert an existing Struts application to Spring MVC ?

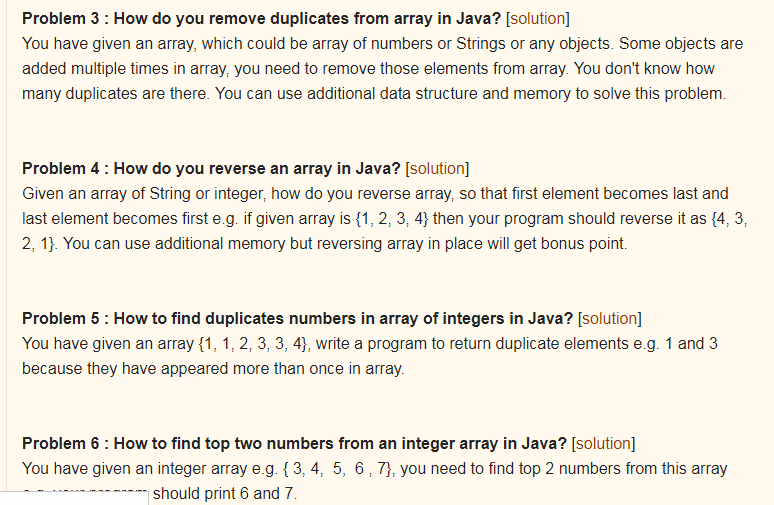
22 Array Concepts Interview Questions Answers in Java 🡺 <http://www.java67.com/2015/07/array-concepts-interview-questions-answers-java.html>

1. 

Above Question is Very Important

1. **Question 4 : Can you use Generics with array?** [[answer](http://javarevisited.blogspot.sg/2011/09/generics-java-example-tutorial.html)]  
   No, you cannot use Generic with array, that's why sometime List is better choice over array in Java.

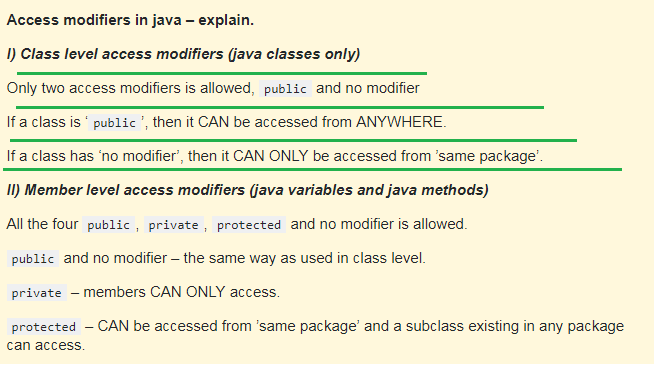
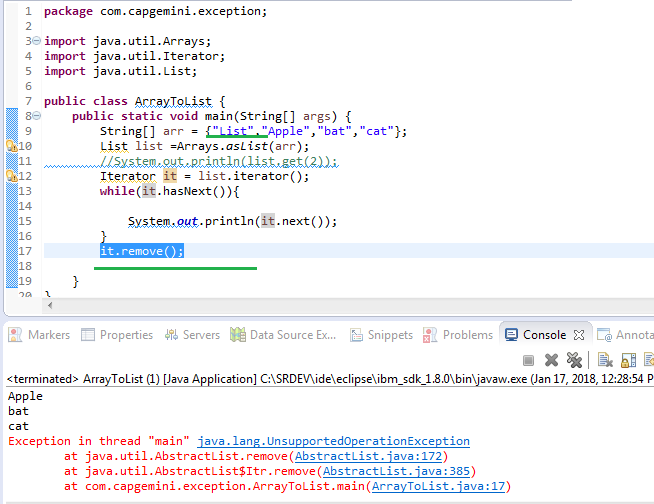
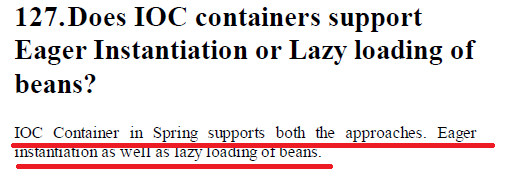
Array Based Programs

1. Problem 2 : How do you find all pair whose sum is equal to given number from integer array in Java?
2. Problem 2 : **How do you find all pair whose sum is equal to given number from integer array in Java?** [[solution](http://javarevisited.blogspot.sg/2014/08/how-to-find-all-pairs-in-array-of-integers-whose-sum-equal-given-number-java.html)]
3.   
     
     
   16/1/2018
4. Can we have more than one class @ted with @Configuration?
5. Not only the add but also remove is not possible with the array converted into list

# Stack overflow 🡺[**Why are interface variables static and final by default?**](https://stackoverflow.com/questions/2430756/why-are-interface-variables-static-and-final-by-default)

Since interface doesn't have a direct object, the only way to access them is by using a class/interface and hence that is why if interface variable exists, it should be static otherwise it wont be accessible at all to outside world. Now since it is static, it can hold only one value and any classes that implements it can change it and hence it will be all mess.

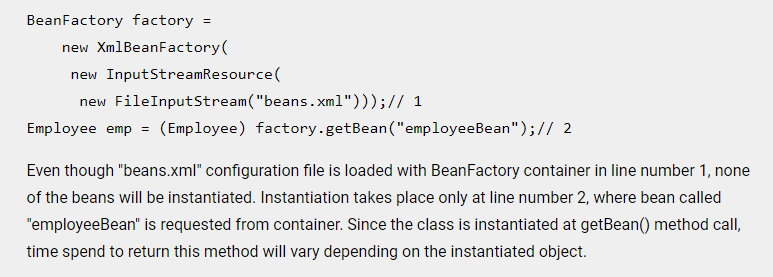
Hence if at all there is an interface variable, it will be implicitly static, final and obviously public!!!

1. **A constructor can have any of the Access specifiers like public, protected, private , none but unlike methods constructor can take up only the access specifier, therefore constructor cannot be abstract, final, static, synchronized, native**
2. 
3. **SEE THE BELOW IMAGE ITERATOR HAS REMOVED THE Object but after removing it had thrown the “UnSupportedOperation” Exception**;
4. Single-Page Applications (SPAs) are Web apps that load a single HTML page and dynamically update that page as the user interacts with the app. SPAs use AJAX and HTML5 to create fluid and responsive Web apps, without constant page reloads. However, this means much of the work happens on the client side, in JavaScript.
5. REGEXP\_SUNSTR()
6. 

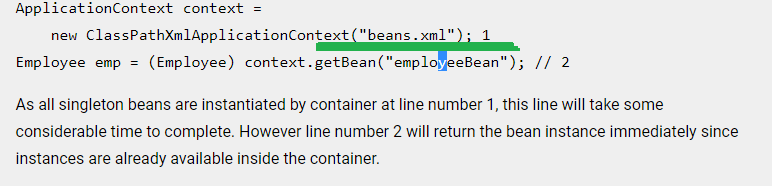
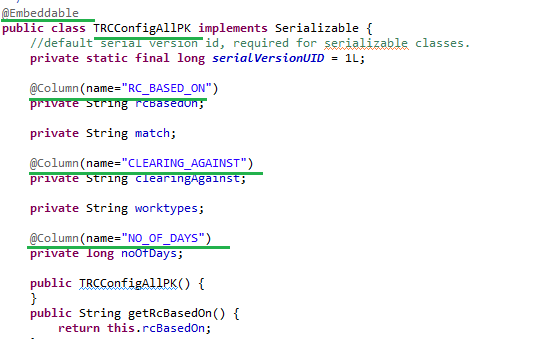
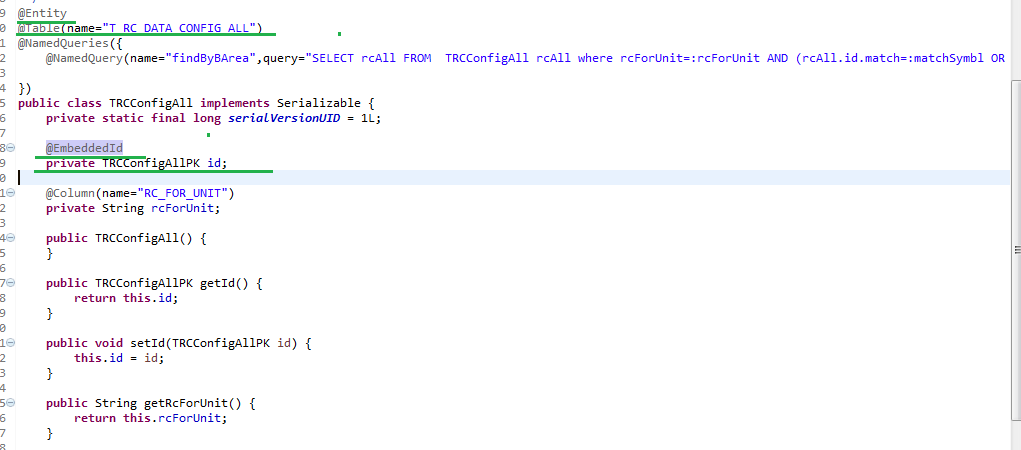
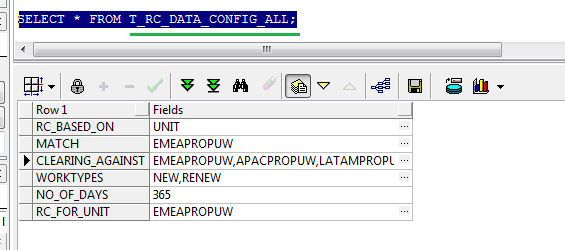
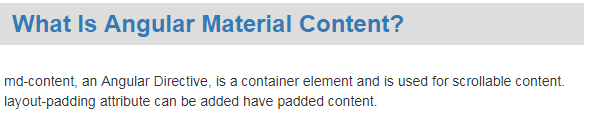
Explanation 🡺

1. Spring framework can instantiate and bind (called loading) related Java objects (called beans) according to a given configuration
2. An XML file can easily be used to define these bindings.
3. Spring framework supports two different types of loading methods; lazy loading and pre-loading respectively managed by BeanFactory and ApplicationContext containers.

## Lazy Loading

1. A bean is loaded only when an instance of that Java class is requested by any other method or a class. org.springframework.beans.factory.BeanFactory (and subclasses) container loads beans lazily.
2. 

## Pre-loading

1. All beans are instantiated as soon as the spring configuration is loaded by a container. org.springframework.context.ApplicationContext container follows pre-loading methodology.
2. 
3. **How will you call a stored procedure in Hibernate? 🡺 needed information on this 🡺 and also check how we had called in CNA**
4. Let’s see an example for how EmbeddedId works 🡺
5. A class which is going to act as a @EmbeddedId field
6. A class which has @EmbeddedId and one of its own field called “ **RC\_FOR\_UNIT** ” 🡺
7. NOW LETS SEE THE TABLE STRUCTURE 🡺
8. See here an interface is having abstract keyword🡺
9.  🡺 needed more information on this
10. **how will you handle errors in Angular2 application**
11. Explain the life cycle hooks of Angular 2 application
12. 