Java Interview Questions

### What is the difference between heap dump and thread dump.

Heap dump - Collection of objects that are in memory (JVM) and useful to analyse OOM situations

Thread dump - Shows what each thread in a process is doing at a given point in time along with the stack trace. These are useful to troubleshoot slow running of your application

Core dump - O/S level dump file which has O/S level info in addition to the heap dump.

Java VisualVM can be used to take heap and thread dumps

What is the difference between JDK, JRE and JVM

JDK – Java development kit. Responsible to develop and run the java application. So, it means JDK = JRE + development tools

JRE – Java runtime environment – responsible to run the application with the help of JVM. JRE = JVM + lib.

JVM = this is the main process

Class path – class file location is given in this. Java compiler and JVM will use it

Path – These is where we tell OS where all the executable binary is located. Like javac in bin folder.

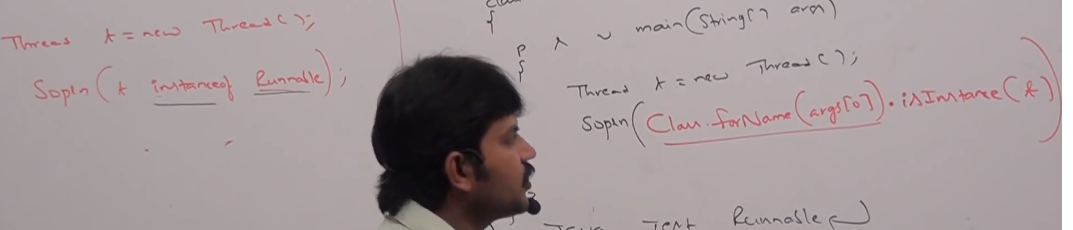
Web Server – This is responsible to run web related technologies like servlet/JSP/HTML etc.

Application Servers – These can run web + all J2ee technologies so app server is like web server + EE technologies.

Difference between instanceof and isinstance().

When we already know the type then we use instanceof

When we don’t know the type (means when it is passes on runtime ) we use isinstance()

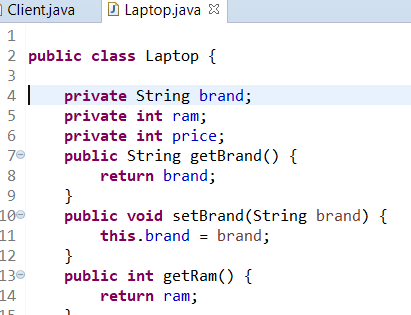


Comparable and Comparator

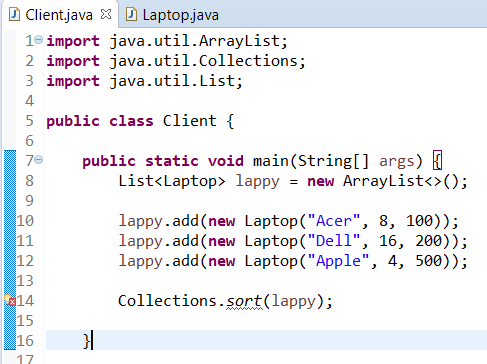
Comparable (For sorting the collection)– Comparable interface comes when we must **sort** some collection.

A collection (of custom type say employee) cannot be sorted until we tell the two employee objects comparing logic specifically.

IF we have a Laptop class like below

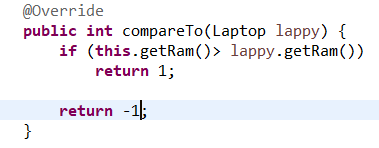


And we want to add several laptops in a list collection and try to sort them in some other class



Here sort method will throw error as it does not know on what basis it needs to sort the laptop objects

So as a thumb rule we always need to extend the custome type class (like employee) with Comparable interface, whose objects we want to sort and implement its compareto(obj o) method like below.



Comparator – Why we need it when we already have comparable

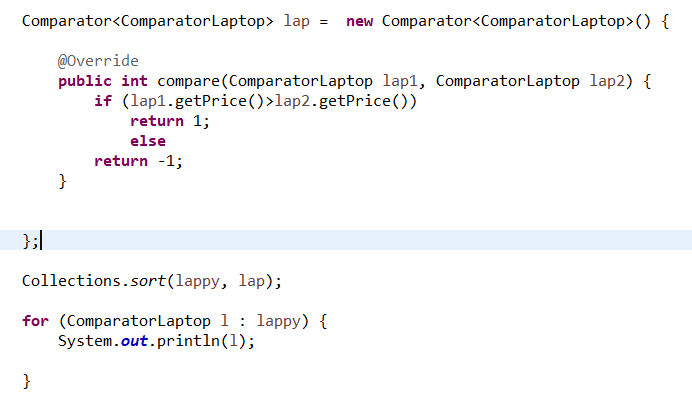
1. **If we cannot extend the already existing class due to the functionality break, we use comparator.**
2. **If we have to change the sorting parameter on the fly (say sorting on the basis on price not on RAM)**

**To implement comparator logic,**

**Collections.sort(List, Comparator obj)** this method takes 2 parameters where first is the list we want to sort and other is the logic of comparison.

This logic is defined below in the overridden compare method of Comparator interface

**Note** – As this is interface so we had to create a anonymous class like below to implement the compare method.



========String in switch statement (Java 7)

Prior to java 7 only premitives types were allowed in switch statement....now we can use string as well in the switch

The switch statement when used with a String uses the equals() method to compare the given expression to each value in the case

statement and is therefore case-sensitive and will throw a NullPointerException if the expression is null

String color ="blue"

switch(color) {

case "red" :

sop

break;

===Binary Literals in Java

Java added a new feature Binary Literal in Java 7.

I allow you to express integral types (byte, short, int, and long) in binary number system.

To specify a binary literal, add the prefix 0b or 0B to the integral value

This feature is very helpful to bit-oriented systems like processors, network protocols and bitmapped hardware device.

Early the programmer used to transform from binary to decimal/hexadecimal and vice versa.

Using this feature will remove this transformation and chances of error will be less in this conversion

byte b1 = 0b101; // Using b0, The b can be lower or upper case

byte b2 = 0B101; // Using B0

System.out.println("----------Binary Literal in Byte----------------");

System.out.println("b1 = "+b1);

System.out.println("b2 = "+b2);

======== try-with-resources (Java 7)

Prior to java 7, we used to use finally block just to close the resourses.(All the IO objects act as resources and need to be closed)

Now we can create the object of that resource as parameter in try block...and once the try block is executed ...resource will be closed.

public class Java7ResourceManagement {

public static void main(String[] args) {

try (BufferedReader br = new BufferedReader(new FileReader(

"C:\\journaldev.txt"))) {

System.out.println(br.readLine());

} catch (IOException e) {

e.printStackTrace();

======== Caching Multiple Exceptions by single catch (Java 7)

Rather than catching multiple exceptions in different catch block. We can club them into one like below

catch(IOException | SQLException ex){

logger.error(ex);

throw new MyException(ex.getMessage());

}

=========Underscores in Numeric Literals (Java 7)

In Java 7 we can put \_ between two numbers of primitive types like int long float

This is convineint put the scenario like credit card numbers

public static void main(String[] args) {

long l = 5454\_5454\_543\_5l;

System.out.println(l);

}

Output - 545454545435

Jenkins Interview Questions

**Q1. What is Jenkins?**

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project

**Use Case**

 First, a developer commits the code to the source code repository. Meanwhile, the Jenkins server checks the repository at regular intervals for changes.

 Soon after a commit occurs, the Jenkins server detects the changes that have occurred in the source code repository. Jenkins will pull those changes and will start preparing a new build.

 If the build fails, then the concerned team will be notified.

 If built is successful, then Jenkins deploys the built in the test server.

 After testing, Jenkins generates a feedback and then notifies the developers about the build and test results.

 It will continue to check the source code repository for changes made in the source code and the whole process keeps on repeating

**Q2. What are the benefits of using Jenkins?**

 At integration stage, build failures are cached.

 For each change in the source code an automatic build report notification is generated.

 To notify developers about build report success or failure, it is integrated with LDAP mail server.

 Achieves continuous integration agile development and test-driven development.

 With simple steps, maven release project is automated.

 Easy tracking of bugs at early stage in development environment than production

**Q3.** **What are the pre-requisites for using Jenkins?**

 A source code repository which is accessible, for instance, a Git repository.

 A working build script, e.g., a Maven script, checked into the repository

**Q4.** **Mention some of the useful plugins in Jenkins?**

* Maven 2 project
* Git
* Amazon EC2
* HTML publisher
* Copy artifact
* Join
* Green Balls

**Q5.** **Mention what are the commands you can use to start Jenkins manually?**

systemctl status Jenkins

**Q6. Explain how to create a backup and copy files in Jenkins?**

Answer to this question is really direct.

To create a backup all you need to do is to periodically back up your JENKINS\_HOME directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. To create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory

**Q7** **Explain how you can deploy a custom build of a core plugin?**

Below are the steps to deploy a custom build of a core plugin:

* Stop Jenkins.
* Copy the custom HPI to **$Jenkins\_Home/plugins**.
* Delete the previously expanded plugin directory.
* Make an empty file called **<plugin>.hpi.pinned**.
* Start Jenkins

**Q8.** **What are the various ways in which build can be scheduled in Jenkins ?**

You can schedule a build in Jenkins in the following ways:

* By source code management commits
* After completion of other builds
* Can be scheduled to run at specified time ( crons )
* Manual Build Requests

**Q9.** **What is the difference between Maven, Ant and Jenkins?**

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

**Q10. What are the two components Jenkins is mainly integrated with?**

According to me Jenkins is mainly integrated with the following:

* Version Control system like GIT,SVN.
* Build tools like Apache Maven.

**Explain what is continuous integration?**

In software development, when multiple developers or teams are working on different segments of same web application, we need to perform integration test by integrating all modules.  In order to do that an automated process for each piece of code is performed on daily bases so that all your code get tested.

**How can you setup Jenkins jobs?**

Follow these steps:

* Select new item from the menu.
* After that enter a name for the job and select free-style job.
* Then click OK to create new job in Jenkins.
* The next page enables you to configure your job.

**Aamir’s interview Questions**

**Interview script for jenkins.**

I have used jenkins in my previous projects as my main Deployment tool. I have used it with Maven , Anisble, and my custom scripts with Jenkins. A typical example of a jenkins build would be that the code would get checked out to a common workspace directory. THe next step in the build would be to compile the code using the maven command line interface. Finally the artifact which was created used to be deployed to the target machines. I have also then written custom jenkisn jobs in shell to do database refreshes across envionments, adhoc nodejs builds, and pipeline creation for deployments. I have got good experience running master slave jenkins servers and configuing build executors in the jenkins front. I have also done rolling upgrades to jekins with proper backup and rollback stratergy.

**Q1-What kind of CI-CD have you done in Jenkins tell me about it?**

A1- I have configured multiple jobs in jenkins and connected with the build pipeline plugin. I have done this for java and web apps (html). I have used both plugins and conf management tools like ansible and chef for deployment.

We had freestyle builds which used to trigger downstream builds in sequence and send out notifications to various stake holders for respective build status messages. Artifacts were pushed to a nexus/central repository. Final deployment was achieved using ansible in most cases.

**Q2- How did u configure authentication in jenkins?**

A1 - So if the team size is less than 10 we would ideally use the jenkins own database with matrix authentication plugin . If the team size is more than 10 then i would integrate it with LDAP so that we can manage roles easily.

**Q3- How do you scale up jenkins**

A3:- you can scale jenkins using two ways. The first one is vertical scaling wherein you can increase the machine size lets say from t2.medium to m4.large.

You can also add jenkins slaves to the master so that they can share the load of the master by runnign some builds.

**Q4-how to backup jenkins?**

A4- You can have a python boto3 script which takes a backup of the machine daily and removed last 7 days image once todays image is success fully created.

Second method is to take daily rsync backup of the jenkins workspace and copy it to some other server ,preferably in a different region or availabilty zone

**Q5- What version of jenkins are you using.?**

Jenkins ver. 2.60.3

**Q6:- What all plugin have you used in jenkins**

git plugin

github plugin

PMD

Cobertura

jUnit

maven

Build pipeline view plugin

HTML plugin

Archive artifacts plugin

Publish PMD reports plugin

Email extension plugin

Matrix base authentication plugin

Matrix based project authentication plugin

Git plugin

jUnit real time tester plugin

Shell Script Interview Questions

**What is the difference between soft and hard links?**

Soft links are link to the file name and can reside on different filesytem as well; however hard links are link to the inode of the file and has to be on the same filesytem as that of the file. Deleting the orginal file makes the soft link inactive (broken link) but does not affect the hard link (Hard link will still access a copy of the file)

**What is the significance of $#, $ and so on as below**

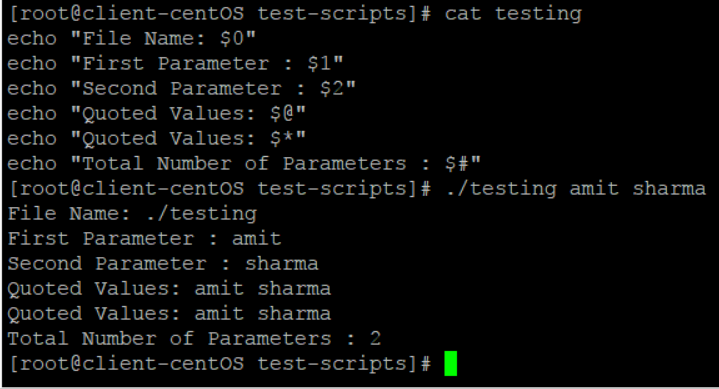
$0 - The filename of the current script.

Echo $# - shows the count of the arguments passed to the script

echo $? – this tells the status of the last command executed (0 means successfully executed and other number means last command was not successful)

$! - The process number of the last background command

echo $$ - This tell the PID of the current shell



**What is the difference between $$ and $!?**

$$ gives the process id of the currently executing process whereas $! shows the process id of the process that recently went into background

**How will you connect to a database server from linux?**

**What are the 3 standard streams in Linux?**

0 – Standard Input  
1 – Standard Output  
2 – Standard Error

**How to set an array in Linux?**

A=(element1 element2 element3 …. elementn)

**Write down the syntax of “for “ loop**

**Write the syntax for “if”**

# Important Misc commands

**man -k command**

**pwdx PID** (it tells the working directory of PID)

**Sticky bit ( if it is on then only creator of file can delete it no matter what the permission is)**

To add sticky bit

chmod 1777 filename

to remove sticky bit

chmod 0777 filename

**How to compress and decompress a file**

To tar a file or folder

*tar -cvf file\_name (this c means create, v means verbose, and f means file)*

untar

*tar -xvf file\_name (here x means extract)*

**to gzip a tar file (compress)**

*gzip file\_name.tar*

**to decompress**

*gunzip file\_name.tar.gz*

**Important** - **here tar only group the files into a single file and does not actually decrease the size(keeps sames size as the actual size)**

**when we do the gzip of the tar file, this actually compress the file)**

**How to do tar and gz in a single command**

**Compress**

tar -zcf file\_name ( this converst the file into file\_name.tgz (which means file\_name.tar.gz)

**To decompress**

tar -zxvf file\_name

## Vi Editor Command

dd => delete the current line. If number precedes dd, then those many lines will be deleted. e.g. 10dd will delete 10 lines from current position.  
cc => change entire line  
yy => yank/copy the line in buffer. It can be placed in the file using p option. If number precedes yy, then those many lines will be copied in buffer. e.g. 10yy will copy 10 lines in buffer, which can be put into file anywhere using p.  
cw => change word.  
ce => change word.

:g/^$/d => delete all the blank lines

0 or ^ => to reach the beginning of line

$ => takes you the end of the line

gg => to reach the beginning of the file

G=> to reach end of file  
ZZ => save changes and exit the file.

## PS command

Ps command and top command are same.ps is just the snapshot at one time while top always gets update with refreshed.

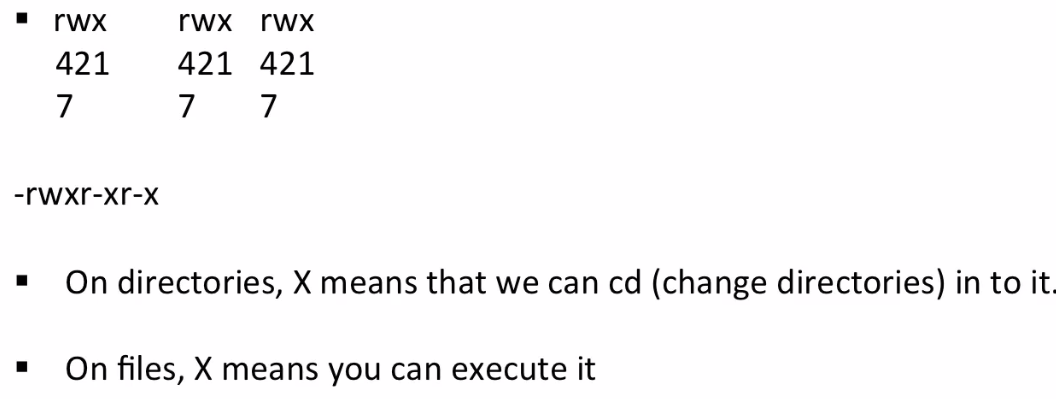
ps aux – this gives all the processes with users

top (this command can also be used to kill some process by hitting k and process id)

## File Permissions

**drwxrwxrwx** 4 root root 96 Sep 16 16:19 file\_comp

First character represents the directory or file



**Symbolic links (sim link)**

Ln –s actual\_file link\_name

Version Control GIT Interview questions

**1)      What is GIT?**

GIT is a distributed version control system and source code management (SCM) system

**2)      What is a repository in GIT?**

A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

**3)      What is the command you can use to write a commit message?**

The command that is used to write a commit message is “git commit –a”.  The –a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use “git add<file>” before git commit –a if new files need to be committed for the first time.

**4)      What is the difference between GIT and SVN?**

The difference between GIT and SVN is

a)      Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.

b)      GIT does not support ‘commits’ across multiple branches or tags.  Subversion allows the creation of folders at any location in the repository layout.

c)        Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

**5)      What are the advantages of using GIT?**

a)      Data redundancy and replication

b)      High availability

**9)      What is “Staging Area” or “Index” in GIT?**

Before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.

**10)   What is GIT stash?**

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory.  So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

**11)   What is GIT stash drop?**

When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command.  It will remove the last added stash item by default, and it can also remove a specific item if you include as an argument.

**12)   How will you know in GIT if a branch has been already merged into master?**

Git branch—merged lists the branches that have been merged into the current branch

Git branch—-no merged lists the branches that have not been merged

**13)   What is the function of git clone?**

The git clone command creates a copy of an existing Git repository.  To get the copy of a central repository, ‘cloning’  is the most common way used by programmers

**14)   What is the function of ‘git config’?**

The ‘git config’ command is a convenient way to set configuration options for your Git installation.  Behaviour of a repository, user info, preferences etc. can be defined through this command

**15) Difference between git pull and git fetch**

Git pull = git fetch + git merge

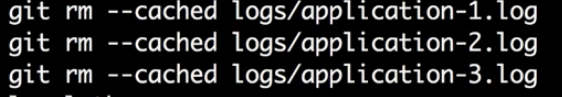
Git fetch :- it takes the commit from the remote repo and keep them on the local repo without merging them. While pull command merge them as well.

Do a git fetch in dev branch then do git diff origin/master (note in this master branch file was updated by other developer)

Now do the git merge origin/master (now this will sync the changes in your dev branch from master branch)

**16) Git RM command**

This command is used to remove the files from local and staging area.



After the above, now do the git commit and push.

**17)   To delete a branch what is the command that is used?**

Once your development branch is merged into the main branch, you don’t need

development branch.  To delete a branch use, the command “git branch –d [head]”.

**18)   What is the difference between ‘git remote’ and ‘git clone’?**

‘git remote add’  just creates an entry in your git config that specifies a name for a particular URL.  While, ‘git clone’ creates a new git repository by copying and existing one located at the URI.

**19)   What is Subgit? Why to use Subgit?**

‘Subgit’ is a tool for a smooth, stress-free SVN to Git migration.  Subgit is a solution for a company -wide migration from SVN to Git that is:

a)      It is much better than git-svn

b)      No requirement to change the infrastructure that is already placed

c)       Allows to use all git and all sub-version features

**20)   What is the difference between the ‘git diff ’and ‘git status’?**

‘git diff’ is similar to ‘git status’, but it shows the differences between various commits and also between the working directory and index.

**21)   What is the function of ‘git checkout’ in git?**

A ‘git checkout’ command is used to update directories or specific files in your working tree with those from another branch without merging it in the whole branch.

**22) In Git how do you revert a commit that has already been pushed and made public?**

There can be two answers to this question and make sure that you include both because any of the below options can be used depending on the situation:

Remove or fix the bad file in a new commit and push it to the remote repository. This is the most natural way to fix an error. Once you have made necessary changes to the file, commit it to the remote repository for that I will use  
git commit -m “commit message”

Create a new commit that undoes all changes that were made in the bad commit.to do this I will use a command  
git revert <name of bad commit>

**23) What is Git rebase and how can it be used to resolve conflicts in a feature branch before merge?**

According to me you should start by saying git rebase is a command which will merge another branch into the branch where you are currently working, and move all of the local commits that are ahead of the rebased branch to the top of the history on that branch