* **Topics**
* **Question**
* **Puzzle**
* **Java Versions**
* **Java**
* [**Threads In General**](#_Threads_In_General)
* **Future Objects**
* [**SubFutureObjects**](#_SubFutureObjects)
* **Callable Objects**

# Topics To Learn

Core Java

OOP Concepts

Exception Handling

Collections

Serialization

UML Notations

Java Memory Management

Garbage Collection

stack/heap/Ram/CPU Cache/CPU registers

Data Structures

Create generic implementaion of a DS

Searching

Sorting

Threads Executor framework

Java Architecture and JVM

Build Management ANT/Maven

Interview Question

Interview Puzzle

Class Diagram / UML

HackerRank

Complexity

Class Loader and Class https://docs.oracle.com/javase/8/docs/technotes/tools/findingclasses.html#bootclass

Boot Strap

JVM

Lamda and method reference

Interface with method

Stream class and APIs https://winterbe.com/posts/2014/07/31/java8-stream-tutorial-examples/

Enum

Inner Class

annonymous Class

Runtime Class

Inner Classes

# Git

## Branch

$ git branch -d branch\_name

$ git branch -D branch\_name

$ git push origin --delete <branch\_name>

deletes the remote branch

Revisit

https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes

:Showing Your Remotes

$ git remote

origin

To see which remote servers you have configured, you can run the git remote command.It lists the shortnames of each remote handle you’ve specified.

If you’ve cloned your repository, you should at least see origin,that is the default name Git gives to the server you cloned from.

$ git remote -v

origin https://github.com/JamesTharakan/cognitiveLearning.git (fetch)

origin https://github.com/JamesTharakan/cognitiveLearning.git (push)

shows you the URLs that Git has stored for the shortname(origin) to be used when reading and writing to that remote

First, you need to create your branch locally

git checkout -b your\_branch

After that, you can work locally in your branch, when you are ready to share the branch, push it.

The next command pushes the branch to the remote repository origin and tracks it.

git push -origin your\_branch

Your Teammates/colleagues can push to your branch by doing commits and then push explicitly

... work ...

git commit

... work ...

git commit

git push origin HEAD:refs/heads/warRoom

git reset HEAD <filePath>

git log --branches --not --remotes=origin

Shows all commits that are in any of local branches but not in any of remote-tracking branches for origin

(what you have that origin doesn’t).

git checkout -b branchname origin/branchname

Here, by default we are setting the upstream branch, so you will not be facing the mentioned issue.

If we push the changes to a <branch> using "-u" , then all your future pushes will be done to that <branch>

# Question

**Why Strings are Immutable**

So that they can be used in hashtable

so that we can use the string pool safely

What is the complexity of recursive left shift operator

prime number with root n complexity

"1. Sort methodsd in collection

2. add() addAll()

3. remove() removeALL()

4. Retain() clear()

5. For loop in collection

6. NavigationSet, NavigationMap"

"Collection col = new HashSet() is better than HashSet set = new Hashset() because we should code to the interface.

But what if, i want to use the methods that are declared and defined in HashSet

Answer : Type Cast works."

"Why is null check bad ?

Cost of null check ismore ?

Is there a difference between null != someThing and someThing != null"

JavaBean and similarities with the Builder Design pattern

Why Type Casting is bad

"Why to Sync a synchronized lis?

https://docs.oracle.com/javase/7/docs/api/java/util/Collections.html#synchronizedList(java.util.List)

# Puzzle

Recursive function to do substring search

Find the missing card in a deck

Build Amazon filters ,builder pattern

Create a Iterator to iterate custom objects

# CoolConcepts

**Branch Prediction:**

"Processing a sorted array id faster than unSorted.

https://stackoverflow.com/questions/11227809/why-is-processing-a-sorted-array-faster-than-processing-an-unsorted-array/11227902#11227902"

# Collection