**Interview Questions: Set 4**

1. What is the difference between StringBuffer and StringBuilder?

StringBuffer is synchronized. This means that multiple threads cannot call the methods of StringBuffer simultaneously.

StringBuilder is asynchronized. This means that multiple threads can call the methods of StringBuilder simultaneously.

Due to synchronization, StringBuffer is called a thread safe class.

Due to its asynchronous nature, StringBuilder is not a thread safe class.

Due to synchronization, StringBuffer is lot slower than StringBuilder. ￼

Since there is no preliminary check for multiple threads, StringBuilder is a lot faster than StringBuffer.

1. What do you mean by String is immutable?

Means once string object is created, it’s data or state cannot be changed but a new string object is created.

1. Does String s=” something” and String s=new String(“Something”) are same?

No

1. What do you understand by String pool and how it works?

String constant pool is special memory location present inside heap area is used to store string literal.

1. What is thread to you?

Thread is the lightweight process, it is the smallest unit of the process, it always exists within a process that means a single process contain multiple threads

1. What are the different ways to create threads?

There are two diff ways through which we can create threads

1.Using Threads class

2.using Runnable interface

1. What is relationship between Runnable and Thread class?
2. What should you prefer using Runnable or Thread?
3. In what way the threads are getting stored in memory?
4. What is the significance of the start () method?

We can start the threads execution using start () method, when we invoke start method control goes to run method and start executing

1. Can you invoke the run () method explicitly?

No, you cannot directly call run method to start a thread

We can inovoke run () method explicitly but thread will not start in that case

1. Explain the lifecycle of thread.

1.New: when ever thread is created, it is always in the new state.

2.Active: when a thread invoke the start method, it moves from new state to active state,the active state contains the 2 state runnable/running

Runnable: when a thread is ready to run then it moves to runnable state

Running: When the thread gets the CPU, it moves from the runnable to the running state.

3. Blocked or Waiting: Whenever a thread is inactive for a span of time (not permanently) then, either the thread is in the blocked state or is in the waiting state

Terminated: A thread reaches the termination state because of the following reasons

1. What is the thread schedular? Who decides which thread is executed?

A component of Java that decides which thread to run or execute and which thread to wait is called a thread scheduler In Java, a thread is only chosen by a thread scheduler if it is in the runnable state.

1. How to set up the priority for the thread?

using the method setPriority() of class Thread. There are three static variables for thread priority in Java i.e. MIN\_PRIORITY, MAX\_PRIORITY and NORM\_PRIORITY. The values of these variables are 1, 10 and 5 respectively.

1. What is the default thread? and what it its priority?

Default priority of a thread is 5 (NORM\_PRIORITY). The value of MIN\_PRIORITY is 1 and the value of MAX\_PRIORITY is 10.

1. What is intrinsic and extrinsic locks?
2. What are examples of extrinsic locks?
3. What is synchronized methods and synchronized block?
4. What is the difference between the normal synchronized method and the static synchronized method?
5. What is the use of join method?

If thread want wait for another thread to complete its task, ten we use join method

1. If I invoke the wait() method without notify() what will happen?
2. How does the wait, notify and notifyAll works?
3. Is multithreading and parallelism is same?
4. What are the types of Executors?
5. What is difference between the submit() and execute() method?
6. What is ForkJoin pool? And how to implement it?
7. Give any practical example of synchronization.
8. What is thread dead lock?

**Exception Handling**

1. What is difference checked and unchecked exception?

**Checked** exceptions that are checked at compile-time called checked exception ex: ClassNotFoundException, IOException, SQLException, ClassNotFoundException etc.

**Unchecked** exceptions that are not checked at compile-time, but they are checked at runtime. ex: ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException, etc.

1. Which is the parent class for Exception hierarchy?

Throwable

1. What is the significance of throws keyword?

**To** **delegate responsibility of exception handling to the caller** **method (JVM or other method)**

The throws keyword is used to declare which exceptions can be thrown from a method, while the throw keyword is used to explicitly throw an exception within a method or block of code. The throws keyword is used in a method signature and declares which exceptions can be thrown from a method

1. Can I use finally without catch?

**Yes, it is not mandatory to use catch block with finally**. You can have to try and finally.

1. Can I use try without catch?

**Yes,** **it is possible to have a try block without a catch block by using a finally block**. As we know, a finally block will always execute even there is an exception occurred in a try block, except System. exit () it will execute always

1. How can we handle the exceptions generically?

T**he try-catch is the simplest method of handling exceptions**

1. How to create user defined or customized exception?

\* If we want to declare our own type of exception then we use user defined exception

\* Throw keyword is best for creating ob. of customized exception

\* If we want handle checked exception then we extend Exception class

We create constructor and display message using super method.

\* If we want handle unchecked exception then we extend Runtimeexception

class UnderAgeException extends Exception {

UnderAgeException() {

super("you are under age");

}

UnderAgeException(String str) {

super(str);

}

}

public class UserDefineException {

public static void main(String[] args) throws UnderAgeException {

int age = 17;

if (age < 18) {

throw new UnderAgeException("you are not eligible");

} else {

System.out.println("you can vote........");

}

}

}

1. How to handle exception?

Using try, throw, throws, catch, finally keyword

1. If we use the finally block without the catch block, what happens if the exception is thrown by the try the block?

If we use finally block without catch block and if exception is occurred in, try block then finally block executed and exception is displayed after executing finally block

1. What is try with resource? Which resources I can use while I work with this concept?
2. What happens the try and finally block execute different Boolean values?

In this case whatever value, you return in finally block will excute whether true or false

1. What happens to try block if the finally block returns some value?

**If finally block returns a** **value, then try and catch blocks may or may not return a value.**

public class ReturnValueFromTryCatchFinally

{

static int methodReturningValue()

{

try

{

//This block may or may not return a value as finally block is returning a value

}

catch (Exception e)

{

//This block may or may not return a value as finally block is returning a value

}

finally

{

return 20;

}

}

public static void main(String[] args)

{

System.out.println(methodReturningValue());

}

}

1. Can we use throws keyword with un-checked exception?

We can declare both types of exceptions using throws clause i.e., checked and unchecked exceptions. But the method calling the given method must handle only checked exceptions. Handling of unchecked exceptions is optional

1. List out different exceptions that you have come across during your entire training.