Java Interview Questions

Question Range: (1-25)

Q: What are the four fundamental concepts of java?

1. Abstraction
2. Inheritance
3. Polymorphism
4. Encapsulation

Q: Why Java is Object Oriented Programming Language?

Java work with object means that every object has its own attribute and its behave exactly same way while it use. In addition to

There are four fundamental concept like :

1. Abstraction
2. Inheritance
3. Polymorphism
4. Encapsulation

Q: Why java is platform independent language?

Platform independence means that we can write and compile the java code in one platform (eg Windows) and can execute the class in any other supported platform eg (Linux,Solaris,etc) because of JVM which actually helps to do that..

Q: Why Java is not pure object oriented programming language?

Java use primitive data type and cannot handle more than one inheritance.

Q: What is javac and jvm, jdk jre?

Javac means java compiler which compile java code

jvm means java virtual machine which execute class file or machine language in order to get result of program

JDK means java development kit which helps to build the program

JRE means java runtime environment which helps to run project.

Q: What is object?

Object is an instance of a class.

Q: What is Class?

Class is a blueprint of its object. It has state and behavior. (by state we mean variable and by behavior we mean method.) its describe state and behavior of that object.

Q: What is the base class of all classes?

Object is base class of java.

Few methods of Object class : finalize(), notify(), notifyAll(), equals(), wait(), wait(x), wait(x, y), getClass(), hashcode()

Q: What is the difference between a public and a non-public class?

A public class may be accessed outside of its package. A non-public class may not be accessed outside of its package.

Q: Can a source file contain more than one class declaration?

Yes, a single source file can contain any number of Class declarations but only one of the class can be declared as public.

Q: What is variable?

Variable is name of memory location.example:int a (variable name)=5;

Q: What is local and global variable?

Local variable: Local variables are declared in methods, constructors, or blocks. Global variable declared outside the method inside the class. Global variable also known as Instance variable.

Q: What is datatype?

Data type is a set of data with having predefined characteristics. Data type also gives some space in memory.

There are two kinds of data type:

1: Primitive

Ex: byte, short, Int, long, float, double,character, boolean.

2: Referenced / Non-primitive:

Ex: Any built in class or interface and any class or interface which we created our self when it use to declare a variable that considered as a referenced data type like: String, Integer are built in class in java

Class A {

A x;//Here A also referenced data type

String s=“ hi”; //here string is built class

}

Q: How to define a constant variable in Java?

The variable should be declared as static and final. So only one copy of the variable exists for all instances of the class and the value can't be changed also.

Example:

static final int MAX_LENGTH = 50; is an example for constant.

Q: What are wrapper classes?

Ans: Java provides specialized classes corresponding to each of the primitive data types. These are called wrapper classes.

They are example: Integer, Character, Double, Boolean etc.

Q: What is class variable?

When global variable declared as a static keyword is called class variable

Q: Difference between instance and class variable?

When you declare a variable with static keyword in a class level that called class variable.

When you declare a variable in a class level without static keyword is called instance variable.

Q: What is default value for different data type?

int ---à0

String -----ànull

Boolean --àfalse

Double ----à0.0

Q: What is the default value of an object reference declared as an instance variable?
The default value will be null unless we define it explicitly.

Q: What is a transient variable?
Transient variable is a variable that may not be serialized.

Q: What is System.out.println?

Ans: System is a class, out is a variable, print is a method.

Q: What is method?

Collection of statements which grouped together to perform an operation.

Q: Should a main() method be compulsorily declared in all java classes?

It is not required for all classes but we need a class with main() method at least once to execute our project.

Q: What is the return type of the main() method?

Main() method doesn't return anything hence declared void.

Q: Why is the main() method declared static?

main() is called by the JVM even before the instantiate of the class hence it is declared as static.

Question Range: (26-50)

Q: What is the argument of main() method?

main() method accepts an array of String object as argument.

Q: What is a package?

Package is a collection of related classes and interfaces.

Q: Which package is imported by default?

java.lang package is imported by default even without a package declaration.

Q: Do I need to import java.lang package any time? Why ?
No. It is by default loaded internally by the JVM.

Q: What is java.io?
Java.io is a package in java where you can get FileInputStream and FileOutputStream etc.

Q: What is modifier? How many type of modifier are there and what are those?

A modifier is a keyword placed in a class, method or variable declaration that changes how it operates.

Modifier are two kinds:

1: access modifier: It gives us access level of class, methods, variable. There are some different access level based on package and to create an object or inherited.

access modifier are 4 kind.

1. default
2. public
3. protected
4. private

Creating an object in same package:

public and default protected can access. private cannot access.

By inheritance in same package:

all access modifier can access except private.

By creating an object in different package:

Only public is accessible

By inheritance in different package:

Only public and protected is accessible

How to access private variable:

By using getter setter method(getter mean read,setter mean write)

example:

```
private int a=8;

 int getA(){

return a;

 }

 int setA(int x){

a=x;

 return a;

 }
```

//in class label:

in top class only public and default can be used and protected, private is not accessible. But in inner class all access modifier is possible.

1. **Non-access modifier:**

Non access modifier are three kinds:

1. final
2. static
3. abstract

Example:

1. **Final:**

If you declare a class as a final, you cannot inheritance but you can create an object. if you declare variable as a final you just can use it but you cannot change. if you declare method as a final you cannot override.

1. **static:**

you cannot declare static in a top class level.

if you declare variable as a static that will work with change value and you can access that with class name. if you declare method as a static you

can overload that method but you cannot override. You can call that with class name and only can take static property.

1. **abstract:**

If you declare class as abstract, you cannot create object but u can inherit. abstract class can take abstract and non-abstract method.

Q: What modifiers may be used with a top-level class?
A top-level or regular class may be public, abstract, or final.

Q: What does it mean that a method or field is "static"?
Static variable means is called class variable.Static variable always work with change value. Static variable u can call with class name means without creating an object. Static methods can be referenced with the name of the class rather than the name of a particular object of the class (though that works too). Static method can be overload but u cannot override.

Q: Can a top class declared as private or protected?

Not possible. But inner class possible.

Q: What is the purpose of declaring a variable as final?

A final variable's value can't be changed. final variables should be initialized before using them.

Q: What is the impact of declaring a method as final?

A method declared as final can't be overridden. A sub-class can't have the same method signature with a different implementation.

Q: I don't want my class to be inherited by any other class. What should i do?

You have to declared your class as final because final class cannot be inherited.

Q: Can you give few examples of final classes defined in Java API?

jString, Math are final classes.

Q: Let's say you have a class which implement a interface but class did not implements all methods then is that possible you can create an object of that class?

No because regular class cannot take unimplemented methods so that you have to declared that class as abstract. As we know abstract class cannot be instantiated.

Q: Can a method inside Interface to be declared as final?

No, not possible. Public and abstract are the only applicable modifiers for method declaration in an interface.

Q: Can a class be declared as static?

We cannot declare top level class as static, but only inner class can be declared static.

Example:

```
public class Test

{

static class InnerClass

{

//is called inner class
```

}

}

Q: When will you define a method as static and what is restriction?

When a method needs to be accessed even before the creation of the object of the class then we should declare the method as static. A static method should not refer to instance variables without creating an instance and cannot use "this" operator to refer the instance.

Q: What is the importance of static variable?

static variables are class level variables where all objects of the class refer to the same variable. If one object changes the value then the change gets reflected in all the objects and can be called by class name.

Q: Can we declare a static variable inside a method?

Static variables are class level variables and they can't be declared inside a method. If declared, the class will not compile.

Q: What is an Abstract Class and what is it's purpose?

A Class which doesn't provide complete implementation is defined as an abstract class. Abstract classes enforce abstraction and cannot create an object of that class.

Q: Can an abstract class be declared final?

No

Q: What is use of abstract variable?
Variables can't be declared as abstract.

Q: What is an abstract method?
An abstract method is a method which is unimplemented.

Q: What is the difference between a static and a non-static inner class?
A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

Q: What is the difference between inner class and nested class?
Nested classes are divided into two categories: static and non-static. Nested classes that are declared static are simply called static nested classes. Non-static nested classes are called inner classes.

Question Range: (51-75)

Q: Can a abstract class be defined without any abstract methods?

Yes, it's possible. This is basically to avoid instance creation of the class.

Q: What is inheritance? Is that possible java can inherit more than one class?

Inheritance is the process by which one object acquires the properties of another object Inheritance is a mechanism wherein a new class is derived from existing class (oop concept).a class cannot inherit more than one

class.

Q: What is casting?
Casting really means is taking an Object of one particular type and “turning it into” another Object type. This process is called **casting**. There are two types of casting, casting between primitive numeric types and casting between object references. Casting between numeric types is used to convert larger values, such as double values, to smaller values, such as byte values. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.

Q: Which object oriented Concept is achieved by using overloading and overriding?
Polymorphism.

Q: What is difference between method overloading and overriding?

Overloading: Overloading happen in a same class. Same method name with different parameter (size/type). Access modifier and return type may be changed. Compile time polymorphism.

Overriding: Overriding happens between parent class and child class. The body will be changed but the access modifier, method name, return type and parameter cannot be changed. Run time polymorphism.

Q: Where do you use overload and overriding in in WebDriver?

Overload: In order to handle frame there are three method with same name (frame()) With different parameter, this is the example of method overloading.

Override: In webdriver interface there are some unimplemented method such as:

get()
getCurrentUrl();
getTitle()
findElements()
findElement()
getPageSource()
close()
quit()

above all methods overridden by FirefoxDriver, ChromeDriver, SafariDriver class

Q: If you create a object it showing an error what could be the reason?
Mainly it could be Constructor parameter did not match

Q: What is constructor? When u need constructor? Can you overload or override a constructor?

Ans: Constructor is a block of code which execute at the time of object creation and it construct an object. To construct an object means let's say you want to change a value of a reference then you can use constructor. Constructor name always same as class name. You can have constructor as many as you want in a class. But you have to change parameter type or parameter size. To change the parameter size or type. You can overload a constructor but cannot be override in java.

Q: What is the difference between a constructor and a method?
A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator.

A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the reference**.**

Q: Does a class inherit the constructors of its super class?
A class does not inherit constructors from any of its super classes.

Q: When does the compiler supply a default constructor for a class?
The compiler supplies a default constructor for a class if no other constructors are provided.

Q: How are this() and super() used with constructors?
this() is used to invoke a constructor of the same class.

super() is used to invoke a superclass constructor.

Q: What is immutable class?

If class is final. String is immutable class. To create a customize immutable class all the property from final class has to be final .

Q: How to divided a string?

```
String s="how are you";
String s1[] =s.split(" ");
```

Q: How to find Sub String from a String?

```
String s="how are you";
s.substring(1,4);
```

Q: How to reverse a String?

```
String sentence="hope you are doing well";
char[] c=sentence.toCharArray();
StringBuffer reverseWord=new StringBuffer();
for(int i=c.length-1;i>=0;i--)
{
        reverseWord.append(c[i]);
}
System.out.println(reverseWord);
```

Q: How to reverse a sentence based on word ex: how are you will be you are how.

```
String sentence=" how are you ";
String word[]=sentence.split(" ");
StringBuffer reverseWord=new StringBuffer();
for(int i=word.length-1;i>=0;i--)
{
      reverseWord.append(word[i]).append(" ");
}System.out.println(reverseWord);
```

Q: What is difference between equals and hashcode methods

Equals method compare between two object if both object is same it returns true if not it returns false

HashCode method compare between two object based on hashCode value if both object value is same it returns true if not it may return true or false both.

Q: .Let's say you have a string(String s="how are you") how can remove all space between word?

```
String s="How are u";
s.replace(" ", "");
```

first parameter takes old character and second parameter takes new character.

Q: How to compare between two string?

```
String s="how";
String s1="are";
```

s.compareTo(s1);

it will return int value -

0- if both are equal

(+) positive int - first object is greater than the second one

(-) negative int - first object is less than the second one

Q: What is the difference between string and string-buffer?

The main difference between String and StringBuffer is string is immutable and StringBuffer is mutable.

Q: What is the difference between stringbuilder and string-buffer?

The main difference between StringBuilder and StringBuffer is StringBuilder is Synchronized and StringBuffer is not synchronized. Also, StringBuffer is faster than StringBuilder

Q: What is the difference between an if statement and a switch statement?
The if statement is used to select among two alternatives. It uses a boolean expression to decide which alternative should be executed. The switch statement is used to select among multiple alternatives. It uses an any expression to determine which alternative should be executed.

Q: What is the difference between a while statement and a do while statement?
A while statement checks condition at the beginning of a loop to see whether the loop will be execute or not. A do while statement checks at the end of a loop to see whether the condition is right or not. The do while statement will always execute the body of a loop at least once.

Q: What is break and continue keyword in loop?

A break statement results in the termination of the statement to which it applies (switch, for, do, or while).
A continue statement is used to end the current loop iteration and return control to the loop statement.

Question Range: (76-100)

Q: Can a for statement loop indefinitely?
Yes, a for statement can loop indefinitely. For example, consider the following: for (; ;);

Q: What is the syntax for ForEach loop / Enhanced Loop?

When we have a variable who has more than one value and using those value if need to do any action bin same page then we should go for for each loop. By default it will increment 1.cannot increment or decrement otherwise.

```
int a [] = {4,6,5,9,8};

for (int x: a)

{

syso(x);

}
```

Q: What is array? Why we need an array?

Array is a container which hold same type of value and fixed size. When you need more than one value in a same variable.

Q: how to get max or minimum number from an array?

```
int a[]={4,76,4,8,9};

int max=a[0];

for(int i=1;i<a.length;i++){
```

```
if(a[i]>max)//if u just say a[i]<max tahole
minimum number paben.
{
max=a[i];
}
}
syso(max)
```

Q: How to get an array as a descending order or ascending ?

```
int a[]={6,8,7,9,4};
Arrays. sort(a);
for(int i=0;i<a.length();i++)
{
 syso(a[i]);//ascending order
}
for(int i=a.length-1;i>=0;i--)
{
syso(a[i]);//descending order
}
```

Q: How to copy one array to another array?

```
int a[]={6,8,7,9,4};
int b[]=new int[a.length];
for(int i=0;i<a.length();i++)
{
b[i]=a[i];
}
```

Q: How to get second highest number?

```
public int secondLargest() {
int a[]={5,8,9,65,76};
int largest, secondLargest;

 if(a[0] > a[1]) {
   largest = a[0];
   secondLargest = a[1];
   } else {
     largest = a[1];
     secondLargest = a[0];
        }

   for (int i = 2; i < a.length; i++) {
          if((a[i] <= largest) && a[i] >   secondLargest) {
   secondLargest = a[i];
```

```
    }

    if(a[i] > largest) {
     secondLargest = largest;
     largest = a[i];
          }
       }

     System.out.println(secondLargest);
```

Q: How to find duplicate value from an array?

```
public boolean mai() {
int a[]={5,8,9,65,76};
for (int i=0; i<a. length; I++)
{
  for (int j=i+1;j<a.length;j++)
  {
     if(a[i]==a[j])
     {
System.out.println("duplicate value is "+ a[i]);
  return true;
  }
}} return false;
```

Q: Are arrays primitive data types?

No, In Java Arrays are objects.

Q: How to declared double dimensions array?

```
int [][] a=new int[2][3];
```

Q: What is Interface?

An **interface** is a reference type in **Java**. It is similar to class but not class. It is a collection of abstract methods(unimplemented method). A class implements an **interface**, thereby inheriting the abstract methods of the **interface**. Along with abstract methods, an **interface** may also contain constants, default methods, static methods.

Q: What is interface? why we need interface?

Ans: Interface is bit like a class but not class; except we can only declare methods and variables in Interface, we cannot actually implement the method in Interface. In other words, Interface is basically a collection of abstract/unimplemented method.

When test scenario is available but acceptance criteria and requirement specifications are not finalized.
Interface features:

Class implements the interface
One class can implement more than one interface at a time.

Interface extends more than one interface.
Interface cannot implement another interface.
Interface centralizes all classes.

Q: Can an Interface extend another Interface?

Yes, an Interface can inherit another Interface, for that matter an Interface can extend more than one Interface.

Q: Why is an Interface be able to extend more than one Interface but a Class can't extend more than one Class?

Basically Java doesn't allow multiple inheritance, so a Class is restricted to extend only one Class. But an Interface is a pure abstraction model and doesn't have inheritance hierarchy like classes (do remember that the base class of all classes is Object). So an Interface is allowed to extend more than one Interface.

Q: Can an Interface be declared as a final?

Ans: No.

Q: Can a class be defined inside an Interface?
Yes it's possible.

Q: Can an Interface be defined inside a class?
Yes it's possible.

Q: What modifiers are allowed for methods in an Interface?
Only public and abstract modifiers are allowed for methods in interfaces.

Q: What is an abstraction and abstract class?

Abstraction is the process of abstraction in Java is used to hide certain details and only show the essential features of the object. In other words, it deals with the outside view of an object .

Abstract classes are **classes** that contain one or more **abstract** methods. An **abstract** method is a method that is declared, but contains no implementation. **Abstract classes** may not be instantiated, and require subclasses to provide implementations for the **abstract** methods.

Q :What does it mean that a method or class is abstract?
An abstract class cannot be instantiated. Abstract methods may only be included in abstract classes. However, an abstract class is not required to have any abstract methods, though most of them do. Each subclass of an abstract class must override the abstract methods of its superclasses or it also should be declared abstract.

Q: Can an abstract class be final?
An abstract class must not be declared as final.

Q: How to execute constructor of Abstract class?
We need to extend the abstract class, where we have to create a constructor-through this constructor by the help of keyword ‘super’ we can reach parent class. Then we have to create object in another class then the constructor of extended class will be executed; this is how the constructor of Abstract class will be executed.

Q: What is difference between interface and abstract?

Abstract class	**Interface**
1) Abstract class can **have abstract and non-abstract** methods.	Interface can have **only abstract** methods.
2) Abstract class **doesn't support multiple inheritance**.	Interface **supports multiple inheritance**.
3) Abstract class **can have final, non-final, static and non-static variables**.	Interface has **only static and final variables**.
4) Abstract class **can have static methods, main method and constructor**.	Interface **can't have static methods, main method or constructor**.

5) Abstract class **can provide the implementation of interface**.	Interface **can't provide the implementation of abstract class**.
6) The **abstract keyword** is used to declare abstract class.	The **interface keyword** is used to declare interface.

7) **Example:**
public abstract class Shape{
public abstract void draw();
}
	Example:
	public interface Drawable{
	void draw();
	}

8) If you add new method to abstract class, you can provide default implementation of it. So you don't need to change your current code	If you add new method to interface, you have to change the classes which are implementing that interface
9) Abstract classes are almost same as java classes except you can not instantiate it	Different from normal java class
10) It is faster than interface	Interface is somewhat slower as it takes some time to find implemented method in class

Q: When a class must be abstract class? When we need abstract class?
An abstract class is a class with collections of implemented and unimplemented method, which cannot be instantiated (instantiated means we cannot create object for abstract class)
Even there is only one unimplemented method in a class then that class is known as Abstract class.
When we have to keep a method as unimplemented in a class-
we have to write that method as abstract method, then the whole class will be Abstract class.

Q: What is polymorphism?
Ans: Polymorphism is a concept which allows u to give the ability an object in different form. This relation is called is a relationship with different form means in order to perform this concept you have to inherit the parent class first then u can use only. It include method overload and override concept as well.
There are two types of polymorphism in java:
compile time polymorphism and runtime polymorphism. We can perform polymorphism in java by method overloading and method overriding.

```
class Animal {

}

class Dog extends Animal {

}

class Cow extends Animal {

}

class Smoke {

        Animal a=new Dog():

       Animal a1=new Cow();

}
```

Question Range: (101-125)

Q: What is encapsulation?
Encapsulation is a mechanism that binds together code and the data it manipulates and keeps both safe from outside the class. Its called data hiding.you can give access if you want using getter and setter method.

Q: Difference between encapsulation and private variable?

Data and information hiding are a broader notion, found in computer science & software engineering. It refers to the fact that those part of a computer program that may change must not be accessible from other modules/from clients.

Encapsulation is a term that is found in Object-Oriented paradigm and refers to keeping the data in private fields and modify it only through methods.

Thus **encapsulation may be seen as a way of achieving data hiding in object-oriented systems**.

Q: How to generate random number using java?
int random = (int)(Math.random() * 5000 + 1);

Q: What is regular expression in java?
A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. They can be used to search, edit, or manipulate text and data.
Here is some expression:

\A Matches beginning of string.
\Z Matches end of string. If a newline exists, it matches just before newline.
\z Matches end of string.
\G Matches point where last match finished.
\n Back-reference to capture group number

"n"
\b Matches word boundaries when outside brackets. Matches backspace (0x08) when inside brackets.
\B Matches nonword boundaries.
\n, \t, etc. Matches newlines, carriage returns, tabs, etc.
\Q Escape (quote) all characters up to \E
\E Ends quoting begun with \Q

Q: What is difference between = and == sign?

= will help to assign value in a variable.

== will help to compare two object.

Q: What is Pattern and Matcher class?

A compiled representation of a regular expression. A regular expression, specified as a string, must first be compiled into an instance of this class. The resulting pattern can then be used to create a Matcher object that can match arbitrary character sequences against the regular expression. All of the state involved in performing a match resides in the matcher, so many matchers can share the same pattern. A typical invocation sequence is thus

Pattern p = Pattern.compile("a*b"); Matcher m = p.matcher("aaaaab"); boolean b = m.matches();

Q: What is the % operator?
It is referred to as the remainder operator. It returns the remainder of dividing the first.

Q: **What is the difference between the Boolean & operator and the && operator?**
& is bitwise. **&&** is logical & evaluates both sides of the operation. a&b- it will consider the both side and && evaluates the left side of the operation, A&&B - if it's true, it continues and evaluates the right side.

Q: Difference between == and equals method?

== sign will compare between two object

equals method will compare between what two object contains.

Q: let say there is number from 1 to 100. If number is visible by by 3 print “hi”, if visible by 5 print “hello”, if visible by 3 and 5 print “hi hello” otherwise print number?

```
for(int i = 1;i<=100;i++){


if(i%3==0&&i%5==0){
syso(“hi hello”)
}
else if(i%5==0){
syso(“hello”)
}
if(i%3==0){
syso(“hi”)
}
else{
syso(i)
}
}
```

Q: What is the purpose of garbage collection in Java, and when is it used?
The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources can be reclaimed and reused. A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used

Q: What is the difference between notify and notifyAll ?

notify wakes (any) one thread in the wait set, notifyAll wakes all threads in the waiting set.

Q: What is serialization?
To **serialize** an object means to convert its state to a byte stream so that the byte stream can be reverted back into a copy of the object. A **Java** o bject is **serializable** if its class or any of its superclasses implements either the **java**.io.**Serializable** interface or its subinterface, **java**.io.Externaliz able.

Q: What is the common usage of serialization?
Whenever an object is to be sent over the network, objects need to be serialized. Moreover if the state of an object is to be saved, objects need to be serialized.

Q: What is synchronization and why is it important?
Synchronization is a process which helps for multithreading means project can take more than one thread or request at the same time. If you don't have this concept in a project, you cannot handle more than one request at same time.

When we start two or more threads within a program, there may be a situation when multiple threads try to access the same resource and finally they can produce unforeseen result due to concurrency issues. For example, if multiple threads try to write within a same file then they may corrupt the data because one of the threads can override data or while one thread is opening the same file at the same time another thread might be closing the same file.

So there is a need to synchronize the action of multiple threads and make sure that only one thread can access the resource at a given point in time. This is implemented using a concept called **monitors**. Each object in Java is associated with a monitor, which a thread can lock or unlock. Only one thread at a time may hold a lock on a monitor.

Java programming language provides a very handy way of creating threads and synchronizing their task by using **synchronized** blocks. You keep shared resources within this block

Q: What is exception in java? How to handle exception?
An **exception** is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

Using try catch block and throw and throws clause.

Q :What is the difference between error and exception?
Error means is major problem which developer cannot handle such as system error.

Exception means the problem which developer can handle by using try catch block or throws keyword.

Q: What are runtime exceptions?
Runtime exceptions are those exceptions that are not warn by compiler but it thrown at runtime.

Ex:StackOverflowException ,MemoryoutException ,ArithmaticException

Q: What are Checked and Unchecked Exception?
The exceptions which are checked by compiler for smooth execution of

the programmed at runtime are called checked exception.
Ex: FileNotFoundException, NoSuchElementException etc .

In the Case of checked exceptions compiler will check whether we are handling exception if programmer not handling then we will get compile time error.

The exceptions which are not checked by compiler are called unchecked exception

Ex: ArrayIndexOutOfBoundsException, ArithmaticException, NuulPointException.
In the case of unchecked exception compiler won't check whether programmer handling exception or not.

Q: What are the different ways to handle exceptions?
There are two ways to handle exceptions,
1. By wrapping the desired code in a try block followed by a catch block to catch the exceptions.
2. List the desired exceptions in the throws clause of the method and let the caller of the method handle those exceptions.

Q: Combination of try catch block
How to handle Exception?
By using Try Catch block we can handle exception. Basic syntacs of Try Catch block and combination of Try Catch block are explained below –
First combination:

```
Try{

 Any statement;}
catch(AnyException Class     Any variable name){
syso(“print whatever you want”);}
```

Second Combination:

```
Try{
  Any statement;}
catch (AnyException Class     variable name){

 syso(“print whatever you want”);}
catch (AnyException Class   variable name){

 syso(“print whatever you want”);}
catch (Exception(-parent calss of exception variable name){
 syso(“print whatever you want”);}
```

Third combination:

```
Try{
Any statement;}
catch (AnyException Class     variable name){
  syso(“print whatever you want”);
}catch (AnyException Class     variable name){

 syso(“print whatever you want”)}
catch (Exception(-parent calss of exception variable name){

  syso(“print whatever you want”);}
finally{

   statements;}
```

Fourth combination

```
Try{
Any statement
}finaly{

statements;}
```

** finally will always execute no matter what.

Q: What is the difference between throw and throws in Java?

Ans: **throw** keyword is used to throw our own exception in the program. **throw new** Exception class **throws** is to declare an exception within a method. e.g suppose some programmer defines a method and he knows that it may causes exception, but he/she don't want to handle that exception, then this method defined with the **throws** keyword. But when we call this function, it should be call with **try** and **catch** block to maintain the normal flow of the program

Q: Is it necessary that each try block must be followed by a catch block?
It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block or a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

Q: If I write return at the end of the try block, will the finally block still execute?
Yes even if you write return as the last statement in the try block and no exception occurs, the finally block will execute. The finally block will execute and then the control return.

Q: How does a try statement determine which catch clause should be used to handle an exception?
When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exception is executed. The remaining catch clauses are ignored.

Question Range: (126-150)

Q: what is the difference between final finally and finalize?
As I said earlier final keyword can be used along with variable, method and Class in Java. If you make a variable final, you can not change it's value, it will act like a constant. final variables are initialized at the time of creation except in case of blank final variable which is initialized in Constructor. If you make a method final in Java, you can not override it in sub class . If you make a class final means it can not be sub classed. Making a class final automatically makes all its method final and this is sometime required due to security reason, final keyword also help to write Immutable classes which are critical for designing thread-safe multi-threading system and reducing amount of synchronization. Now let's see What is finally in Java? As I said finally is used for exception handling along with try and catch. As per Java programming language's rule, for exception handling you at least need either catch or finally block. finally block has special advantage over catch that its guaranteed to be executed despite whether Exception is thrown or not, this makes it, an ideal place to close system resource e.g.InputStream or OutputStream, which is required to release scarce file descriptor. Closing streams, network connection, database connection in finally block is good coding practice in Java. By the way from Java 7 you can use try with resource block to close resource automatically. Since finally is guaranteed to be executed on most cases, finally block always execute, except in case of JVM dies i.e. calling System.exit() .
Now let's see What is finalize() method, finalize() is called by Garbage collection thread just before collecting eligible Objects. This is the last chance for object to perform any cleanup but since its not guaranteed that whether finalize() will be called, its bad practice to keep resource till finalize call. Though you can build a safety net on finalize by double checking scarce resources. See 10 points on finalize method to know more about specific points of finalize(). So, final, finally and finalize all are different keyword, they are used for different purpose. only similarity between them is that they are a Java programming language keyword, other than that final, finalize and finally are completely different than each other.

Q: What if System.exit() in try block?

System.**exit exits** the program immediately, bypassing any other code execution (such as finally **blocks**). If you want to **exit** the program after finally **blocks** run, throw an exception instead. If JVM **exits** while the **try** or catch code is being executed e.g. **System**.**exit**() , then the finally **bloc k** may not execute

Q: Do you know what are the current version of all the tools you are using?

Ex: java---1.8 version

Selenium –3.3.0

And so on .Try to know the version which tool are u using.

Q: What are the steps in the JDBC connection?
While making a JDBC connection we go through the following step

```
@Test
public void jdbc() throws ClassNotFoundException {
Connection con = null;
Statement st = null;
ResultSet rs = null;
String url = "jdbc:mysql://localhost:3306/testdb";
String user = "root";
String password = "";
```

```
try {
Class.forName("com.mysql.jdbc.Driver"); // to initialize the driver
con = DriverManager.getConnection(url, user, password); // create connection in to data base
st = con.createStatement(); // to create statement
rs = st.executeQuery("SELECT VERSION()"); // to get result set
if (rs.next()) {
System.out.println(rs.getString(1)); // to print the 1st value from table
}
} catch (SQLException ex) {
Logger lgr = Logger.getLogger(JDBCConnection.class.getName());
lgr.log(Level.SEVERE, ex.getMessage(), ex);
} finally {
try {
if (rs != null) {
rs.close();
}
if (st != null) {
st.close();
}
if (con != null) {
con.close();
}
} catch (SQLException ex) {
System.out.println("......");
}
}
}
```

Sample image of above code:

```
public void jdbc() throws ClassNotFoundException {
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    String url = "jdbc:mysql://localhost:3306/testdb";
    String user = "root";
    String password = "";
    try {
        Class.forName("com.mysql.jdbc.Driver"); // to initialize the driver
        con = DriverManager.getConnection(url, user, password); // create connection in to data base
        st = con.createStatement(); // to create statement
        rs = st.executeQuery("SELECT VERSION()"); // to get result set
        if (rs.next()) {
            System.out.println(rs.getString(1)); // to print the 1st value from table
        }
    } catch (SQLException ex) {
        Logger lgr = Logger.getLogger(JDBCConnection.class.getName());
        lgr.log(Level.SEVERE, ex.getMessage(), ex);
    } finally {
        try {
            if (rs != null) {
                rs.close();
            }
            if (st != null) {
                st.close();
            }
            if (con != null) {
                con.close();
            }
        } catch (SQLException ex) {
            System.out.println("......");
        }
        }
    }
```