

**3. What are the exception handling keyword in java?**

= There are mainly 5 keyword to handle the exception:

**a) try:** The block of code which may be the cause of raise exception that block of code we put with in the try block. (*The "try" keyword is used to specify a block where we should place exception code*). After the “try” block we have to use catch or finally block. That’s mean we can’t use try block along.

**b) Catch:** We use Catch block to handle the exception. It’s work as handler. We can not use catch along. We have to use Catch after the Catch block.

**c) Throw:** Sometime we want to create our own exception object. After the creation of object we throw the object and handle it**. “throw” keyword is used for throw the exception object.**

**d) Throws:** The **"throws"** keyword is used for declare exceptions. It doesn't throw an exception**. It says that there may occur an exception in the method. It is always used with method signature.**

**e) Finally:** The "finally" block is used to execute the important code of the program which must be executed. It is executed whether an exception is handled or not.

## **4. Common Scenarios of Java Exceptions**

There are given some scenarios where unchecked exceptions may occur. They are as follows:

### **a) A scenario where ArithmeticException occurs**

If we divide any number by zero, there occurs an ArithmeticException.

**int** a=50/0;//ArithmeticException

### **b) A scenario where NullPointerException occurs**

If we have a null value in any [variable](https://www.javatpoint.com/java-variables), performing any operation on the variable throws a NullPointerException.

String s=**null**;

System.out.println(s.length());//NullPointerException

### **c) A scenario where NumberFormatException occurs**

The wrong formatting of any value may occur NumberFormatException. Suppose I have a [string](https://www.javatpoint.com/java-string) variable that has characters, converting this variable into digit will occur NumberFormatException.

String s="abc";

**int** i=Integer.parseInt(s);//NumberFormatException

### **d) A scenario where ArrayIndexOutOfBoundsException occurs**

If you are inserting any value in the wrong index, it would result in ArrayIndexOutOfBoundsException as shown below:

**int** a[]=**new** **int**[5];

a[10]=50; //ArrayIndexOutOfBoundsException

### **5. Explain Java Exception Hierarchy?**

= The java.lang.Throwable class is the root class or parent class of Java Exception hierarchy. It has 2 child. A) Exception B)Error.

Now exception can be divided into two parts. Checked and unchecked.

All the exception which is checked at compile time and which parent class is Exception that is comes under Checked exception.

All the exception which is checked at Run time and which parent class is RuntimeException that is comes under unchecked exception.

Error is an abnormal condition which cal not be handled by us. Error mainly comes for hardware failure like **OutOfMemoryError, StackOverflow etc.**



**6. Multiple Catch Blocks:**

A try block can be followed by one or more catch blocks. Each catch block must contain a different exception handler. So, if you have to perform different tasks at the occurrence of different exceptions, use java multi-catch block.

* At a time only one exception occurs and at a time only one catch block is executed.\*\*
* All catch blocks must be ordered from most specific to most general, i.e. catch for ArithmeticException must come before catch for Exception.

**try**{

**int** a[]=**new** **int**[5];

            a[5]=30/0;

        }

**catch**(ArithmeticException e)

       {

              System.out.println("Arithmetic Exception occurs");

        }

**catch**(ArrayIndexOutOfBoundsException e)

         {

              System.out.println("ArrayIndexOutOfBounds Exception occurs");

           }

**catch**(Exception e)

          {

            System.out.println("Parent Exception occurs");

          }

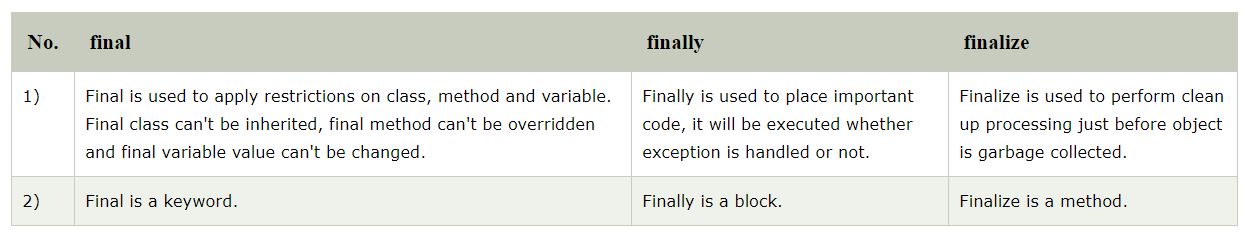
System.out.println("rest of the code");

Here arithmetic exception occure. So in the first catch we use Arithmetic exception as a handler. And at the end we use exception class as a handler. Now if we put the arithmetic execption at last and take the exception class at first then control first move to the Exception class handler but its not move to the most specific exception “ArithmaticException” which is not a example of goo programmer.

**7. Difference Between Throw and Throws:**

|  |  |
| --- | --- |
| **Throw** | **Throws** |
| 1. Java throw keyword is used to explicitly throw an exception. | 1. Java throws keyword is used to declare an exception |
| 2. Throw is used within the function | 2. Throws is used beside the method signature. |
| 3. We can not throw multiple exception | 3. We can declare multiple exception here.  public void method()throws IOException,SQLException |

### **8. What is difference between final, finally and finalize in Java?**



### **9. What is OutOfMemoryError in Java?**

=

OutOfMemoryError in Java is a subclass of java.lang.VirtualMachineError and it’s thrown by JVM when it ran out of heap memory. We can fix this error through java options.

### **10. What are different scenarios causing “Exception in thread main”?**

Some of the common main thread exception scenarios are:

* **Exception in thread main java.lang.UnsupportedClassVersionError**: This exception comes when your java class is compiled from another JDK version and you are trying to run it from another java version.
* **Exception in thread main java.lang.NoClassDefFoundError**: There are two variants of this exception. The first one is where you provide the class full name with .class extension. The second scenario is when Class is not found.
* **Exception in thread main java.lang.NoSuchMethodError: main**: This exception comes when you are trying to run a class that doesn’t have main method.

### **11. What happens when exception is thrown by main method?**

=

When exception is thrown by main() method, Java Runtime terminates the program and print the exception message.

### **12. Can we have an empty catch block?**

= We can have an empty catch block but it’s the example of worst programming. We should never have empty catch block because if the exception is caught by that block, we will have no information about the exception.

\*\*user define exception