**Exceptions:**

There are two types of errors in java.

Compile time errors and Runtime errors.

Compile time errors are errors which will be reported during compilation.

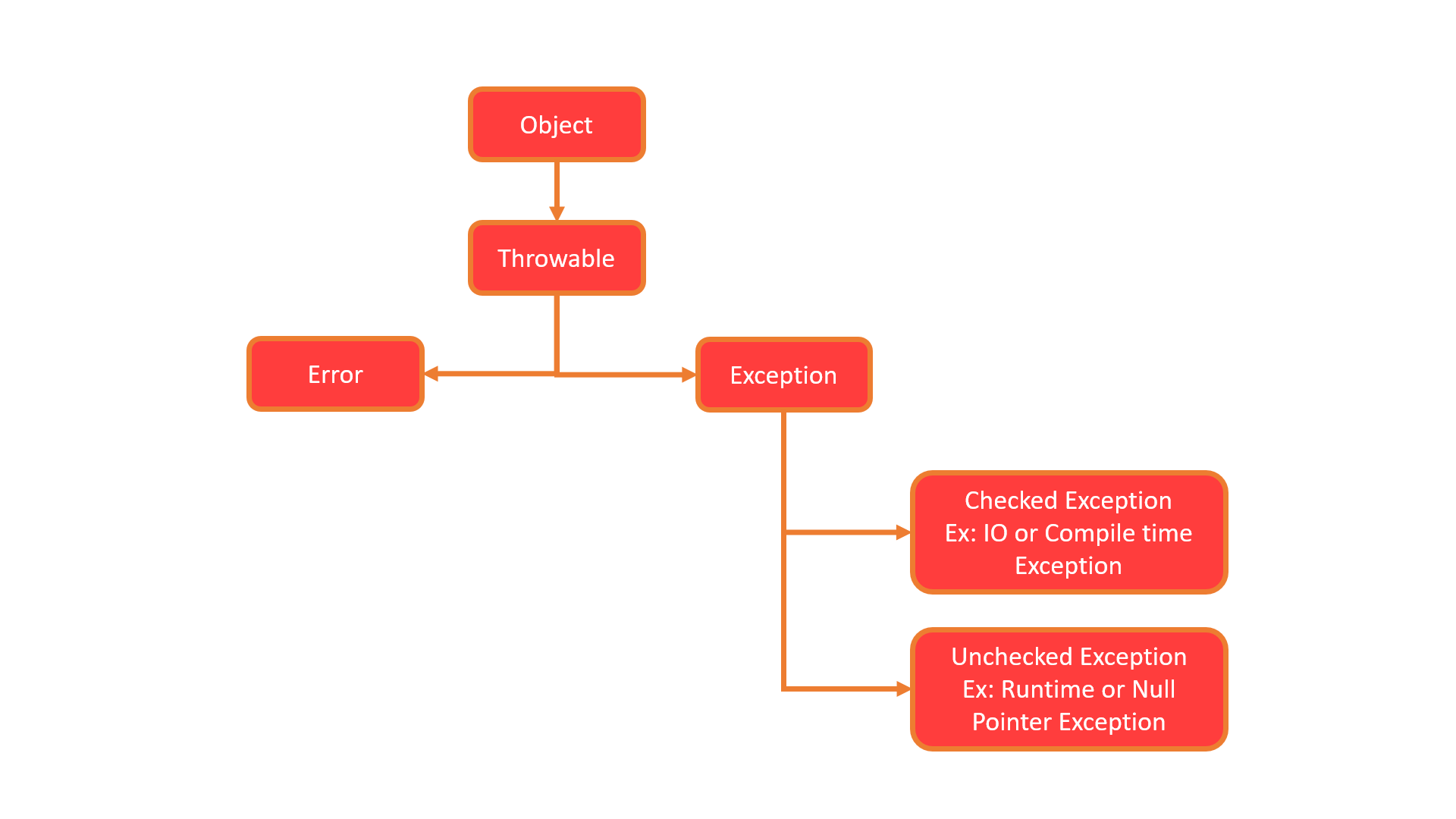
Example:

When you are declaring an integer variable, if you declare “in a” instead of “int a”, this will be a compile time error.

Run time errors are called exceptions.

Example:

Dividing something by zero will throw ArithmaticException.



Error and Exception are sub classes of class Throwable, which is the base class.

**How to handle exceptions in Java:**

Java Exception handling is managed via five keywords – try, catch, finally, throws and throw.

try – catch – finally:

If we think a particular piece of code throws an exception we put it in try block and we write the code to catch the exception in catch block.

Any code that must be executed after a try catch block completes is put in a finally block.

The exceptions we handle using try catch block are called unchecked exceptions.

Ex: ArithmaticException, ArrayIndexOutOfBoundsException etc

Throws:

Sometimes, we know that some methods throw some exceptions and these can be handled by using throws keyword when we define a class. These are checked exceptions.

Ex: IOException, FileNotFoundException, SQLException etc.

Throw:

In java we can define our own exceptions using Throw keyword. These are known as user-defined or custom exceptions.

**How to Handle Exceptions:**

Using Try Catch Block:

Syntax:

try{

Statements;

}

catch(Exception e){

Statements;

}

finally{

Statements;

}

You can use finally block which will be executed irrespective whether an exception raised or not. This is optional.

**Example:**

**package** FPPackage;

**public** **class** StringsDemo {

**public** **static** **void** main(String[] args) {

**int** arr1[] = {10,20,30};

**int** arr2[] = {2,5,10};

**for**(**int** i=0; i<3; i++) {

**for**(**int** j=0; j<3; j++) {

System.***out***.println(arr1[i]/arr2[j]);

}

}

}

}

In the above example we are dividing each element of arr1 with each element of arr2 and printing the results. But if any values of arr2 is 0 it will throw ArithmaticException.

Ex:

**package** FPPackage;

**public** **class** StringsDemo {

**public** **static** **void** main(String[] args) {

**int** arr1[] = {10,20,30};

**int** arr2[] = {2,5,0};

**for**(**int** i=0; i<3; i++) {

**for**(**int** j=0; j<3; j++) {

System.***out***.println(arr1[i]/arr2[j]);

}

}

}

}

It will print 5 and 2 and then it reports arithmetic exception.

We use try catch blocks to handle these kinds of exceptions and we will continue further.