

Exception and checked (compile time) Exception.

Main difference between RuntimeException and checked Exception is that, It is mandatory to provide try catch or try finally block to handle checked Exception and failure to do so will result in compile time error, while in case of RuntimeException this is not mandatory.

Some of the most common Exception like NullPointerException, ArrayIndexOutOfBoundsException are unchecked and they are descended from java.lang.RuntimeException.

Popular example of checked Exceptions are ClassNotFoundException and IOException and that's the reason you need to provide a try catch finally block while performing file operations in Java as many of them throws IOException.

If you I ask my personal opinion, I think Checked Exceptions makes our code code UGLY by adding boiler plate code in for of try-catch finally block.

### 39. Difference between throw and throws in Java?

- throw and throws are two keywords related to Exception feature of Java programming language.
- throw keyword is used to throw an exception explicitly, on the other hand, throws keyword is used to declare an exception which means it works similar to the try--catch block.
- If we see syntax wise than throw is followed by an instance of Exception class throws is followed by exception class names.
- throw new ArithmeticException ("Arithmetic Exception"); throws ArithmeticException;
- throw keyword is used to method body, while throws is used in method signature to declare the exception.

Both of them are two keywords related to Exception feature of Java. As I remember the main difference between throw and throws is in their usage and functionality.

- throws is used in method signature to declare Exception possibly thrown by any method, for example

```
public void shutdown() throws IOException{
    throw new IOException("Unable to shutdown");
}
```

But throw is actually used to throw Exception in Java code.

```
Throw new Exception("is Not able to initialized");
```

In other words; throws keyword cannot be used anywhere exception method signature while throw keyword can be used inside method or static initializer block provided sufficient exception handling.

Oh, I remember one other thing about throw, throw keyword can also be used to break a switch statement without using break keyword

### 40. Difference between Object and Class?

- Class is a blueprint or template which you can create as many objects as you like Object is a member or instance of a class
- Class is declared using class keyword, Object is created through new keyword mainly.

A class is a template for objects. A class defines object properties including a valid range of values, and a default value. A class also describes object behavior. An object is a member or an "instance" of a class and has states and behaviors in which all of its properties have values that you either explicitly define or that are defined by default settings.

Class - A class can be defined as a template/blueprint that describes the behavior/state that the object of its type support.

If we compare them there are many differences but let me tell you some of them which are important to know;

- There are many ways to create object in java such as new keyword, newInstance() method, clone() method, factory method and deserialization. There is only one way to define class in java using class keyword.
- Object is created many times as per requirement. Class is declared once.
- Object is an instance of a class. Class is a blueprint or template from which objects are created.
- Object is a physical entity. Class is a logical entity.