**Access Modifiers**

Access modifiers are nothing but the keywords which set the accessibility of the classes, methods and other variables. They determine which other classes or objects can access the particular or specific member. Java has four types of access modifiers, they are,

* Private
* Default
* Protected
* Public

**Private Access Modifier**

This type of access modifier can be accessible only within the class, not outside the class. A private class is not applicable for top level classes. The private method or variable is accessible only within the class it is declared in and not accessible in other packages or classes.

**Significance :**

* Provides the most restrictive level of access.
* Useful for hiding implementation details and encapsulating the internal state of a class.
* Helps in maintaining the integrity of the class state by preventing external access and modification.

**Default Access Modifier**

This type of access modifier can access only within the same package of different classes. A method or variable with no access modifier is accessible only within its own package.

**Significance :**

* Restricts the visibility to within the package, promoting encapsulation within the package.
* Useful for classes and members that are intended to be used only within a specific package and not exposed to the rest of the application.

**Protected Access Modifier**

This type of access modifier can access within the same package or different package but it should be extended from the parent class. A protected method or variable is accessible within its own package (like the default access modifier), and also accessible in subclasses even if they are in different packages.

**Significance :**

* Provides a way to give subclasses special access to certain methods and variables.
* Useful in inheritance hierarchies to allow derived classes access to parent class methods and fields.

**Public Access Modifier**

The Public class access modifier can be accessible in any type of scenario, regardless of package it resides in. A public method or variable can be accessed from any other class, regardless of the package.

**Significance :**

* Provides the widest scope of accessibility.
* Useful for API methods and classes intended for use by any other classes.

**Exception and Error**

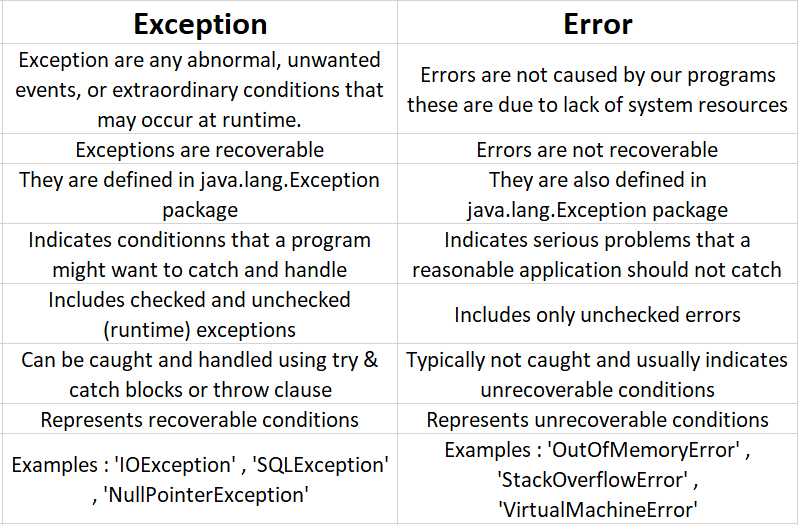
* In Java, both Exception and Error are subclasses of throwable class which means they both can be thrown and caught, but they represent different kinds of problems that can occur during the execution of a program.

* **Exceptions** are conditions that a program can anticipate and handle. They represent issues that arise due to incorrect logic or invalid input, and they can often be recovered from.

* **Exceptions** are for conditions that an application might want to catch and handle, such as invalid input or IO problems. They can be checked (mandatory to handle) or unchecked (runtime exceptions).

* **Errors** are serious issues that are typically beyond the control of the program and cannot be anticipated or handled. They usually indicate severe problems that a reasonable application should not try to catch.

* **Errors** are serious conditions typically outside the control of the application, such as memory issues or JVM problems. They are unchecked and usually indicate situations that the application cannot reasonably be expected to recover from.



**Checked and Unchecked Exception**

Exception is an unwanted event, which occurs during the execution of a program, i.e at run time that disrupts the normal flow of the program’s instructions. There are two types of exceptions, they are,

* Checked Exceptions
* Unchecked Exceptions

Checked Exception :

These are exceptions that must be either caught or declared in the method signature. They represent conditions that an application should anticipate and handle.

Unchecked Exception:

These are exceptions that the compiler does not force you to handle. They typically indicate programming errors that should be fixed in the code.

