**2. Difference Between Exception and Error**

In java both Exception and Error are subclasses of Throwable, but they serve different purposes and are used to handle different types of problem in program.

**Exception** represents conditions that a program might want to catch and handle.

This exception can be handled by java complier.

Examples include IOException, ArrayIndexOutOfBoundsException, and NullPointerException, etc..

Exceptions can be categorized into:

i) Checked Exception

ii) Unchecked Exception

**Checked Exceptions:** These are exceptions that must be either caught or declared in the method where they can be thrown. For example, FileNotFoundException is a checked exception. You handle these exceptions with try-catch blocks or by declaring them in the method signature with the throws keyword.

**Unchecked Exceptions:** These are exceptions that are not required to be caught or declared. They include runtime exceptions like ArithmeticException and IndexOutOfBoundsException. They usually indicate programming bugs, such as logic errors or improper use of an API.

**Error** represents serious problems that a program usually cannot handle. It cannot handle or identified by the java compiler.

Errors are typically used by the JVM to indicate problems that are not meant to be caught or handled by application code. Examples include OutOfMemoryError, StackOverflowError, and VirtualMachineError.

**Use of Exception and Error**

Exceptions to handle predictable problems that your code might encounter, such as invalid user input or file I/O issues., Exceptions allow you to provide meaningful messages or recovery options to handle errors gracefully

Errors are typically not meant to be caught or handled because they often signify critical problems that are beyond the scope of the application to fix. Errors usually indicate that the JVM or system is in a state where recovery is not feasible, such as running out of memory or stack space.