**3. CHECKED and UNCHECKED Exception**

In java Exception is handle by two different ways checked and unchecked exceptions.

Handling exception is one of the important aspects that prevent from event gets terminate in between the program execution.

**CHECKED Exception:**

In general, checked exceptions represent errors outside the control of the program. For example, the constructor of [FileInputStream](https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/io/FileInputStream.html#%3Cinit%3E(java.io.File)) throws FileNotFoundException if the input file does not exist.

Java verifies the checked exception at compile-time of the program. Therefore, we use the throws keyword at beginning of the method so exception is catch at start or we can also use the Try-Catch block to catch the checked exception.

The **throws** keyword is used in the method signature to declare that the method may throw one or more exceptions. This informs the caller of the method that they need to handle these exceptions.

Some of the common checked exception are

**IOException** – throws for input and output exception while reading or writing file fails.

**FileNotFoundException** – throws when an attempt to open a file denoted by file path is failed.

**ClassNotFoundException –** throws when an application tries to load a class by using class.forname() but class is misspelled or missing.

**InterruptedException –** throws when a thread is waiting, sleeping or otherwise occupied, and another thread interrupts it.

**UNCHECKED Exception:**

If a program throws an unchecked exception, it reflects some error inside the program logic. For example, if we divide a number by 0, Java will throw ArithmeticException:

Unchecked exception is only got at run-time of the program, so user must handle the logical termination by implementing in try-block.

**Try-Catch** is the exception handling statement used in java program to catch the unchecked exception at the runtime of the program.

Some of the common Unchecked exception are

**NullPointerException –** when the variable or method need a string value but if we pass null this exception will be catch.

**ArrayIndexOutOfBoundsException –** it occurs we are trying to manipulate the array of index which is not available in the array declaration.

**ArithemeticException** - when a program evaluates an arithemetic operation and it result in some exceptional condition.

**IllegalArgumentException** – when we given the input in the variable or method which is inappropriate argument.