| **Feature** | **Checked Exception** | **Unchecked Exception** |
| --- | --- | --- |
| 📘 **Definition** | Exceptions that are **checked at compile-time**. | Exceptions that are **not checked at compile-time**. |
| 👀 **Compiler Checking** | Compiler **requires** handling (try-catch or throws). | Compiler **does not require** handling. |
| 📂 **Belongs to** | Subclasses of java.lang.Exception **(except RuntimeException)** | Subclasses of java.lang.RuntimeException |
| 🛠️ **Handling Required?** | **Yes**, or the code won't compile. | **No**, optional to handle. |
| 🎯 **Common Examples** | IOException, SQLException, FileNotFoundException | NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException |
| 📍 **Use Case** | Used for **recoverable** conditions. | Used for **programming errors or bugs**. |
| ⛔ **Program Fails At** | **Compile-time** if not handled. | **Runtime** (may crash without warning). |

Throwable

├── Exception → Used for recoverable conditions

└── Error → Used for serious system errors (not meant to be caught)

Exception

├── IOException ← Checked

│ └── FileNotFoundException

├── SQLException ← Checked

├── ClassNotFoundException ← Checked

├── InterruptedException ← Checked

├── RuntimeException ← Unchecked

├── NullPointerException

├── ArithmeticException

├── ArrayIndexOutOfBoundsException

├── NumberFormatException

└── IllegalArgumentException

 ✅ **Checked Exceptions**: Compiler forces you to handle.

 ❌ **Unchecked Exceptions (RuntimeException)**: No need to handle at compile time, but may crash at runtime.

**2. Error (Serious Failures)**

**👉 Represent serious problems that a program usually should not try to handle.**

Error

├── OutOfMemoryError

├── StackOverflowError

├── VirtualMachineError

└── AssertionError

* Not meant to be caught.
* Usually indicates **JVM or system-level problems**.