**BASIC EXCEPTION HANDLING**

* **EXCEPTION**
* EVENT that occurs during the execution of a program that disrupts the normal flow of instruction
* **EXCEPTION HANDLING**
* the PROCESS used to change the normal flow of the code execution if a specified exception occurs.
* **CHECKED EXCEPTIONS**
* occurs during compilation
* cannot be ignored and should be resolved to be able to execute the program
* **EXCEPTION (checked)**

1. ClassNotFoundException
2. IllegalAccessException
3. InstantiationException
4. NoSuchMethodException

* **UNCHECKED EXCEPTION**
* occurs during execution
* also known as RUNTIME EXCEPTION
* ignored at he time of compilation
* **EXCEPTION (unchecked)**

1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. ArrayStoreException
4. InputMismatchException
5. NullPointerException
6. NumberFormatException
7. StringIndexOutOfBoundsException

**CHECKED EXCEPTIONS**

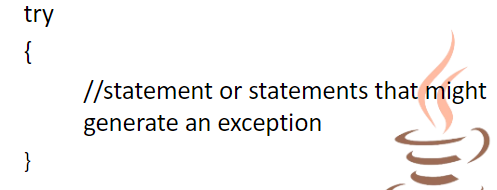
* **ClassNotFoundException**
* the class not found
* **IllegalAccessException**
* access to a class is denied
* **InstantiationException**
* attempt to create an object of an abstract class or an interface
* **NoSuchMethodException**
* a requested method does not exist

**UNCHECKED EXCEPTIONS**

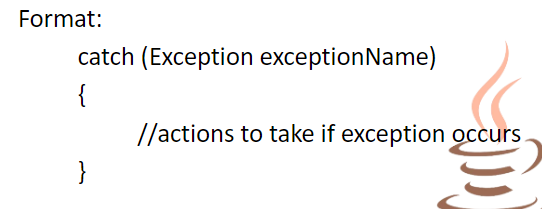
* **ArithmeticException**
* arithmethic error, such as an integer divided by 0
* **ArrayIndexOutOfBoundsException**
* accessing an invalid index of the array
* **ArrayStoreException**
* assigning a value to an array index that does not match the expected data type
* **InputMismatchException**
* entering a value that does not match the expected data type
* **NullPointerException**
* invalid use of null reference
* **NumberFormatException**
* invalid conversation of a string to a numeric format
* **StringIndexOutOfBoundsException**
* accessing an invalid index (character) of a string

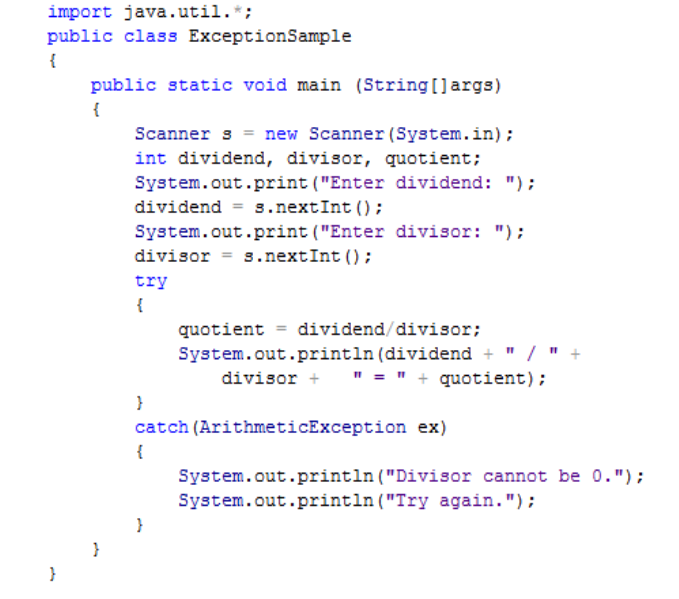
**TRY, CATCH, and FINALLY**

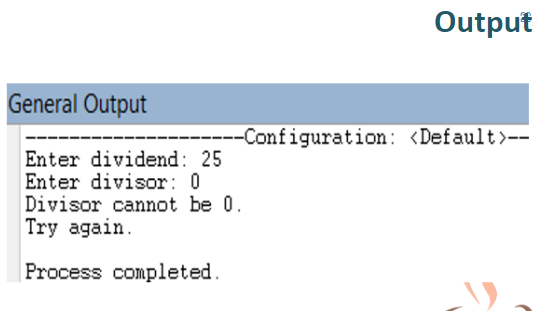
* **TRY BLOCK**
* a block of code that might throw an exception that can be handled by a matching catch block

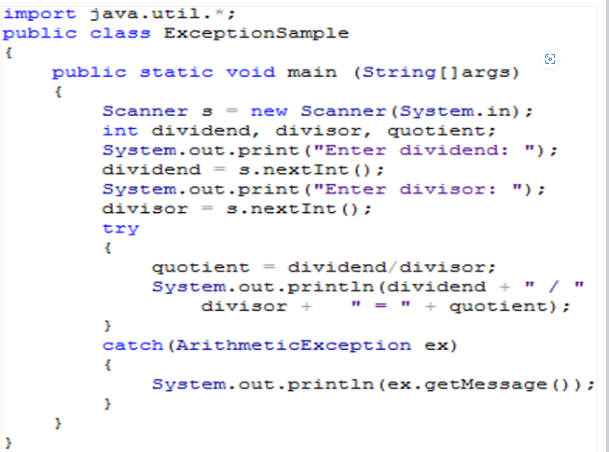


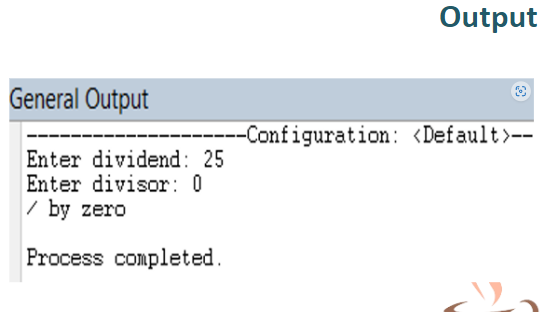
* **CATCH BLOCK**
* a segment of code that can handle an exception that might be thrown by the try block that precedes

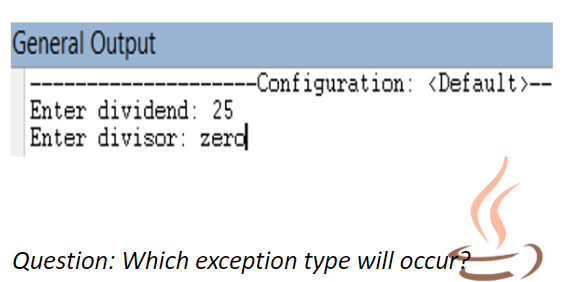


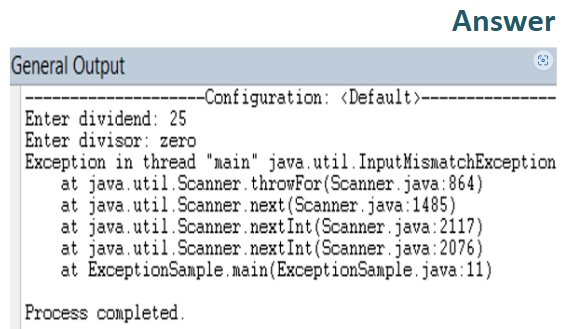


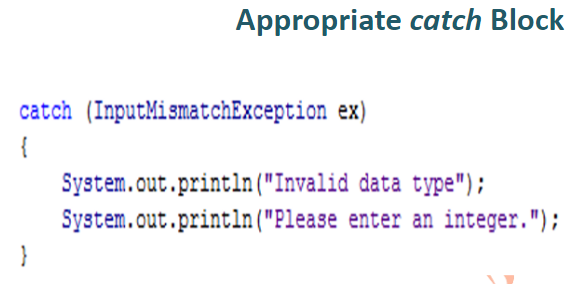


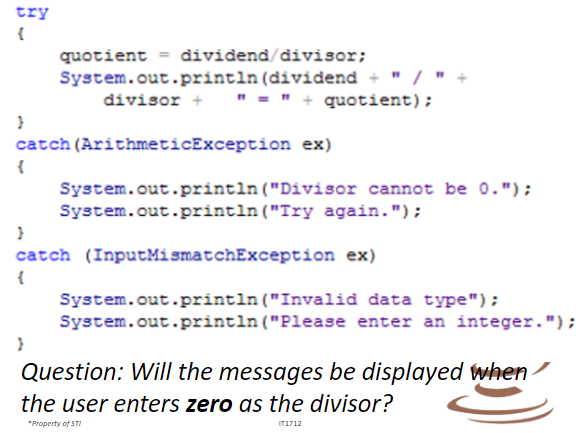


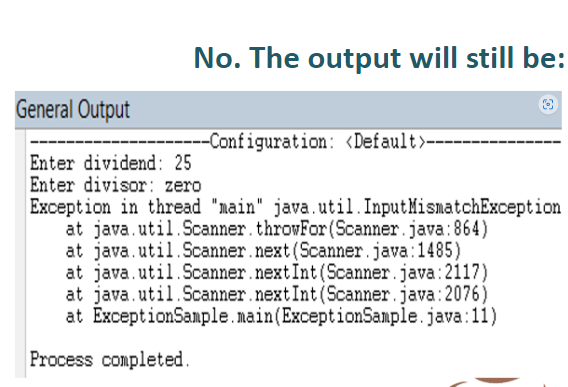


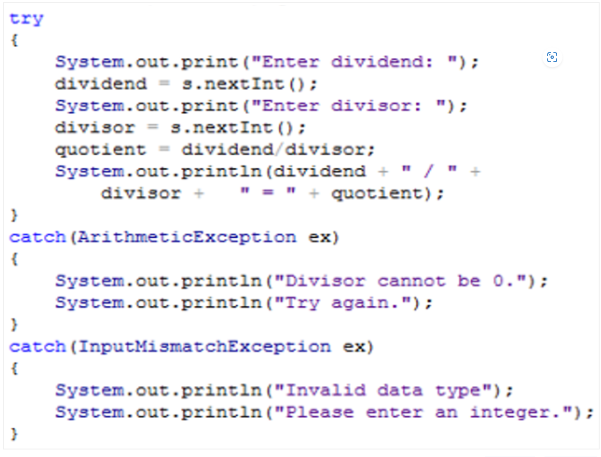


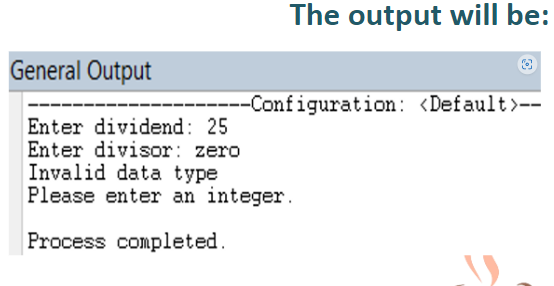


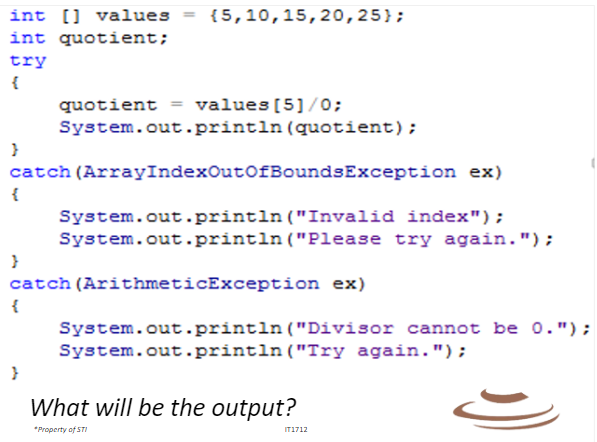


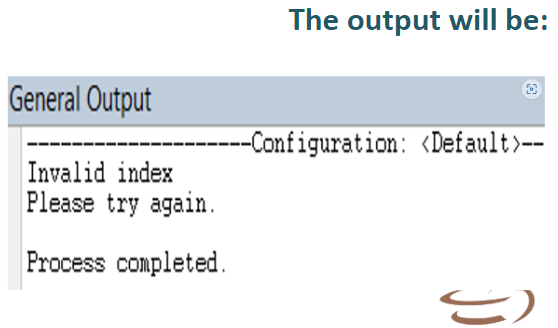






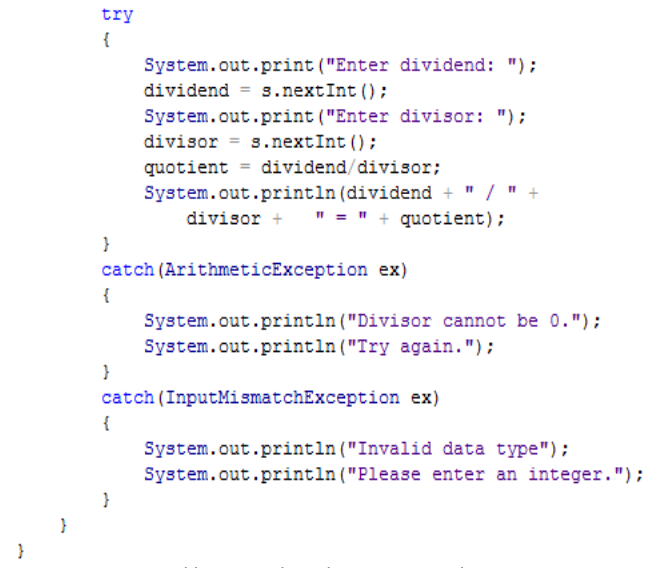




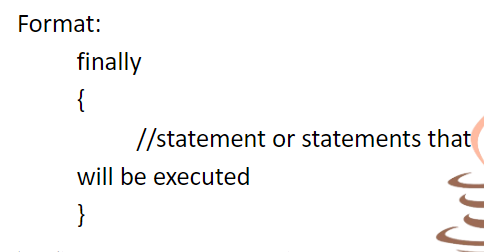


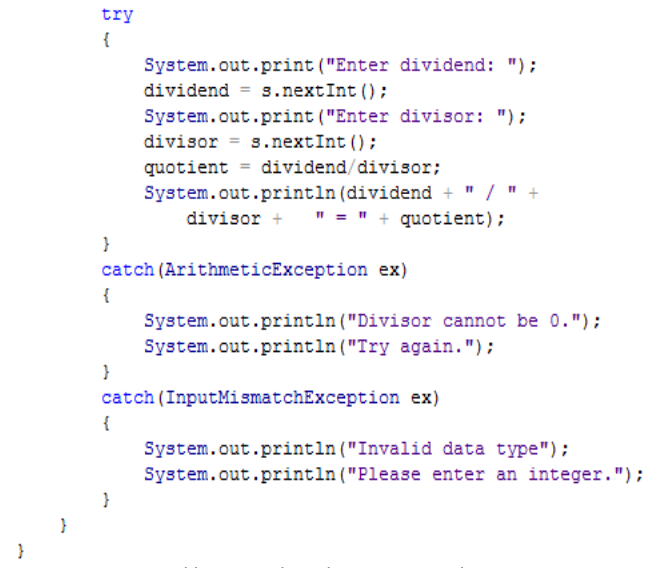
* **getMessage( )**
* method can be used to determine Java's message about the exception.
* Syntax: System.out.println(exceptionName.getMessage( ));

***ONLY ONE (1) TRY BLOCK IS ACCEPTED IN A PROGRAM BUT THERE CAN BE MULTIPLE CATCH BLOCK.***

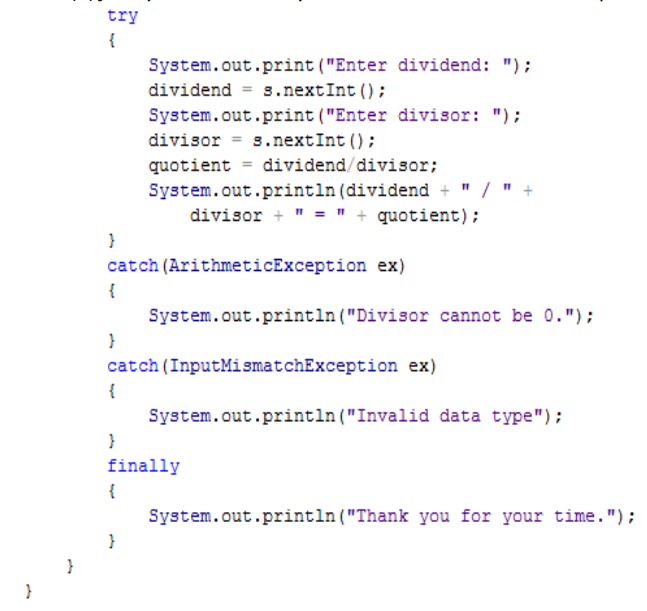
******

* **FINALLY BLOCK**
* contains statements which are executed whether or not an exception is thrown.



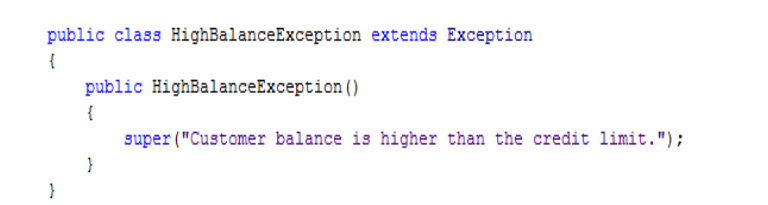
******

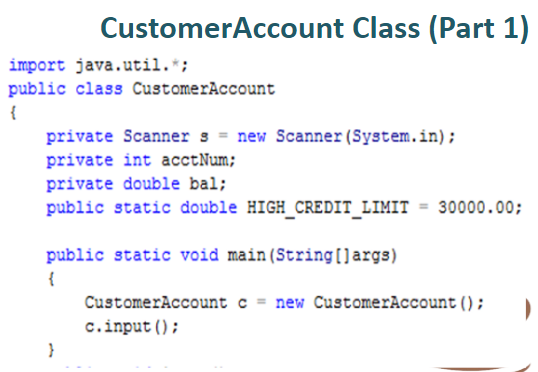
***THERE CAN ONLY BE ONE (1) FINALLY BLOCK AFTER A TRY-CATCH STRUCTURE BUT IT IS NOT REQUIRED***.

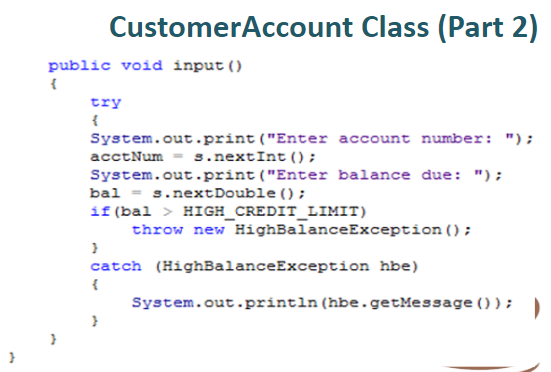


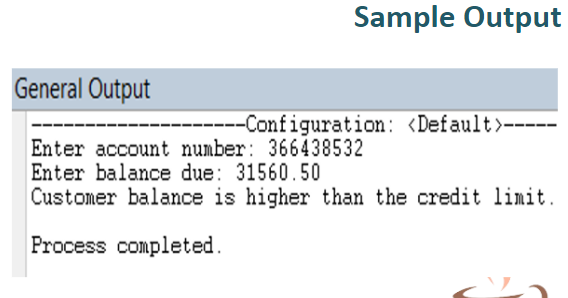
**USER-DEFINED EXCEPTIONS**

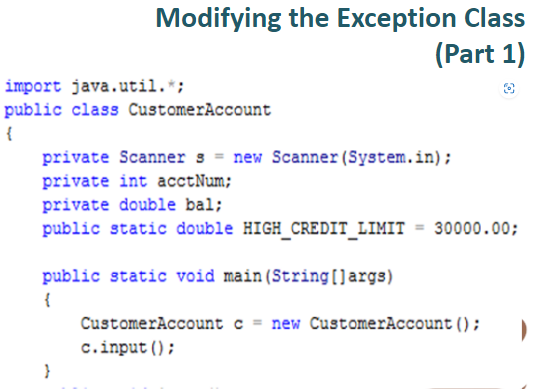
* **USER-DEFINED EXCEPTION**
* created by extending the exception class

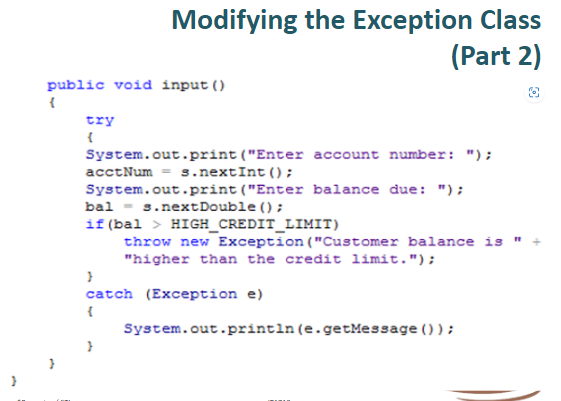












* **THROW STATEMENT**
* sends an exception out of a block or a method so it can be handled elsewhere.

