Exception Handling:

* It is used to handle the runtime errors so that normal flow the code will take place.
* There are 3 types of exceptions:

1. **Checked exception**: all the compile time exceptions are checked exceptions

Ex: IOException, SQLException etc

1. **Unchecked exception**: all runtime exceptions are called unchecked exceptions

Ex: ArithmeticException, ArrayOutOfBoudException, NullPointerException etc

1. **Error**: we cant handled errors Ex: StackOverFlow Error, outOfMemoryError.

Throwable 🡺 it is main Parent Exception class

**Try**: whatever code we assume to get exception that code need to be placed in try block

**Catch**: whatever exception is thrown that exception we need to handled in catch block

**Finally**: whether exception occurs or not but finally block code will execute.

**Throw**: it throws the exceptions

**Throws**: it is used to declare the exception on method which is thrown.

Exceptions are predefined, Custom User defined

Exceptions hierarchy

try{

…

…

}catch(ArithmeticEX ex){

} finally {

..

}

Throw new RuntimeException(“ ”);

1-10 🡺 3 ==? 7 lines

Throwable -🡪 Parent class

1. Exceptions

* Checked exception 🡪 compile time 🡪 Throwable: SQLEXception, IOException
* Unchecked Exception 🡪 Runtime Exception 🡪 ArrayOutOfBoundException, ArithmeticException, NullpointerException.

1. Errors
   1. outOfMemoryError 🡪 Heap memory size increase
   2. stackOverflowError 🡪 primitive data memory
   3. virtualMemoryError

