***Runtime exceptions summary.***

- Every class that extends RuntimeException class is runtime exception.

It is not required to handle(declare) it.

- Checked exceptions.

Every class that extends Exception class is checked exception except RuntimeException.

It is required to handle or declare them.

- Errors.

Every class that extends Error class is error.

Errors are system level failures and we don't handle them or throw them.

--------------------------

How do you handle exceptions?

We handle exceptions by using try catch statement.

try {

// potential code that might throw exception

} catch(TypeOfExceotionWeAreCatching e) {

// hanlde here. Do something

} catch(AnotherTypeOfException e1) {

// hanlde here. Do something

}

- if we are catching parent class exception, it can catch all child class exceptions.

- You cannot put smaller exception after the bigger one in the catch block.

--------------------------

try {

} catch () {

} finally {

// it always runs

// usually, we close our resources inside finally block.

}

--------------------------

Excpetion Declaration

- we can delcare exception in the method declaration.

- we are letting the client code know that our method potentially can throw exception

public void getFile() throws FileNotFoundException {

}

- if method throws checked exception, then client code must handle it or declare it again.

pubic void readContent() {

try {

File file = getFile();

} catch(Exception e) {

e.printStackTrace();

}

// more code

}

pubic void readContent() throws FileNotFoundException {

File file = getFile();

}

------------------------------------

public void openConnectionToDb() throws Exception {

// ...

if (connectionCannotBeDone) {

throw new Exception();

}

}

------------------------------------

What's are the differences between runtime and checked exceptions?

- runtime excpetions are optional to handle and checked exceptions we must handle them.

- All runtime exceptions extends from RuntimeException class and checked exceptions are extends from Exception class(except RuntimeException).

How do you handle exceptions in Java?

- I use try catch.

When do you get NullPointerException?

- when object is null and then when we try to use some properties or methods.

String str = null;

System.out.println(str.length()); // NullPointerException is here

What are the final, finally, and finalize keyword in java?

final:

- if we make our variable final, we cannot change the value(constants)

- if we make our class final, we cannot extend it.

- if we make our method final, we cannot override it.

finally:

- we use finally together with try catch statement. finally block always runs. Usually, we use it to close our resources (file, DB connections).

finalize:

- suggestion to GC for cleanup.

What is the Error in java?

- Usually, they thrown by JVM.

- We don't work with error - we don't handle them, and we don't throw them.

- Examples for error - memory overflow.