

# Exceptions in Java



Create  
Your Future



BUSINESS DEVELOPMENT  
GROUP

# Consider the following points

- Understanding Exceptions
- Hierarchy of Exceptions
- Types of Exceptions
- Handling Exceptions
- Throwing Exceptions



BUSINESS DEVELOPMENT  
GROUP

# Understanding Exceptions

A program can fail for just about any reason:

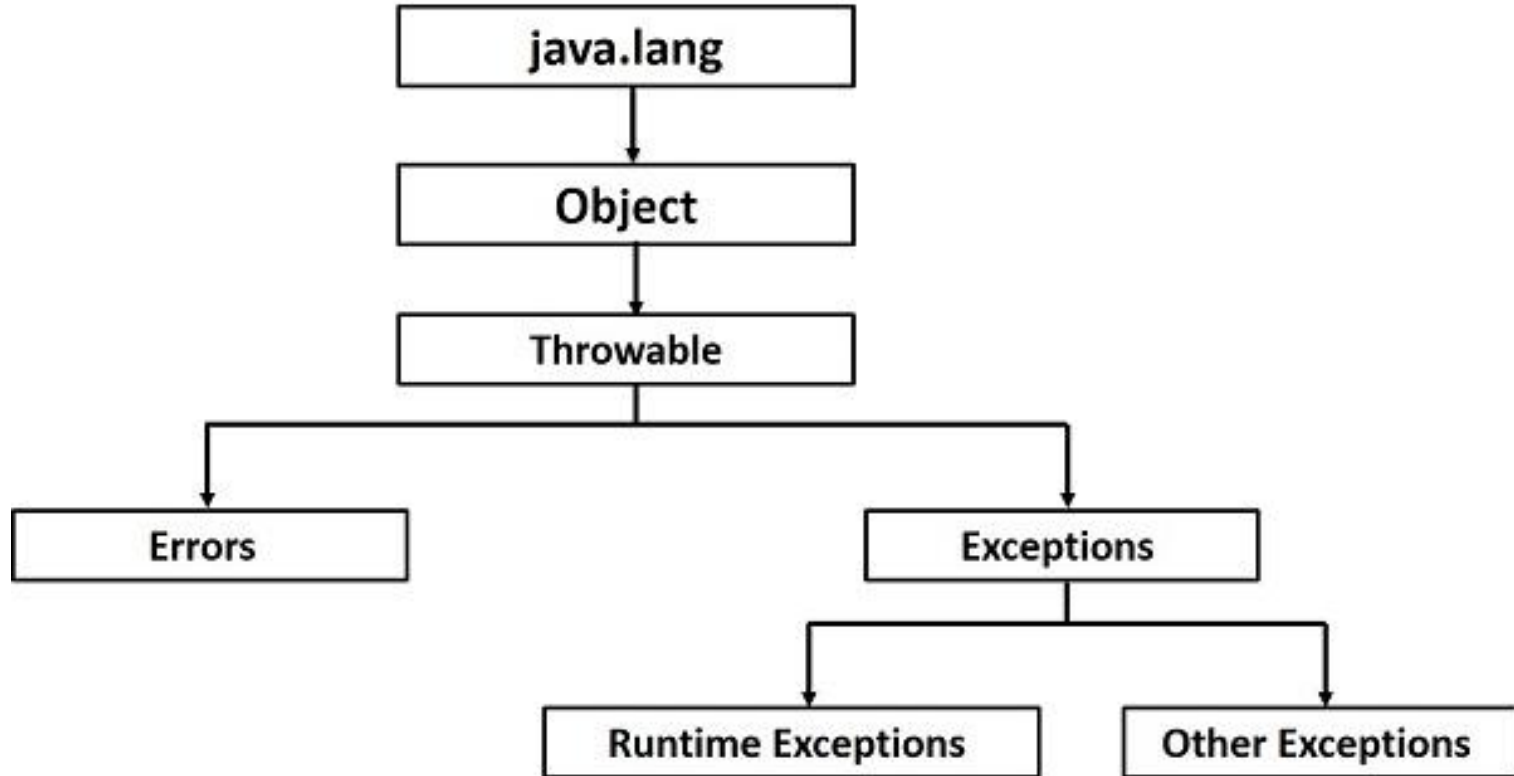
- The code tries to connect to a website, but the Internet connection is down.
- You made a coding mistake and tried to access an invalid index in array.

An **Exception** is Java's way of saying, "I give up, I don't know what to do right now. You deal



BUSINESS DEVELOPMENT  
GROUP

# Hierarchy of





BUSINESS DEVELOPMENT  
GROUP

# Types of Exceptions

Type	How to recognize	Ok for program to catch?	Is program required to handle or declare?
Runtime exception	Subclass of RuntimeException	Yes	No
Checked exception	Subclass of Exception but not subclass of RuntimeException	Yes	Yes
Error	Subclass of Error	No	No



BUSINESS DEVELOPMENT  
GROUP

# Runtime Exceptions

**ArithmeticException** – thrown by the JVM when code attempts to divide by zero.

**ArrayIndexOutOfBoundsException** – thrown by the JVM when code uses an illegal index to access an array.

**ClassCastException** – thrown by the JVM when an attempt is made to cast an exception to a subclass of which it is not an instance.

**IllegalArgumentException** – thrown by the programmer to indicate that a method has been passed an illegal or inappropriate argument.

**NullPointerException** – thrown by the JVM when there is a null reference where an object is required.

**NumberFormatException** – thrown by the programmer when an attempt is made to convert a string to a numeric type but the string doesn't have an



BUSINESS DEVELOPMENT  
GROUP

# Checked Exceptions

**FileNotFoundException** – thrown programmatically when code tries to reference a file that does not exist.

**IOException** – thrown programmatically when there is a problem reading or writing a file.



BUSINESS DEVELOPMENT  
GROUP

# Error

S

**ExceptionInInitializerError** – thrown by the JVM when a static initializer throws an exception and doesn't handle it.

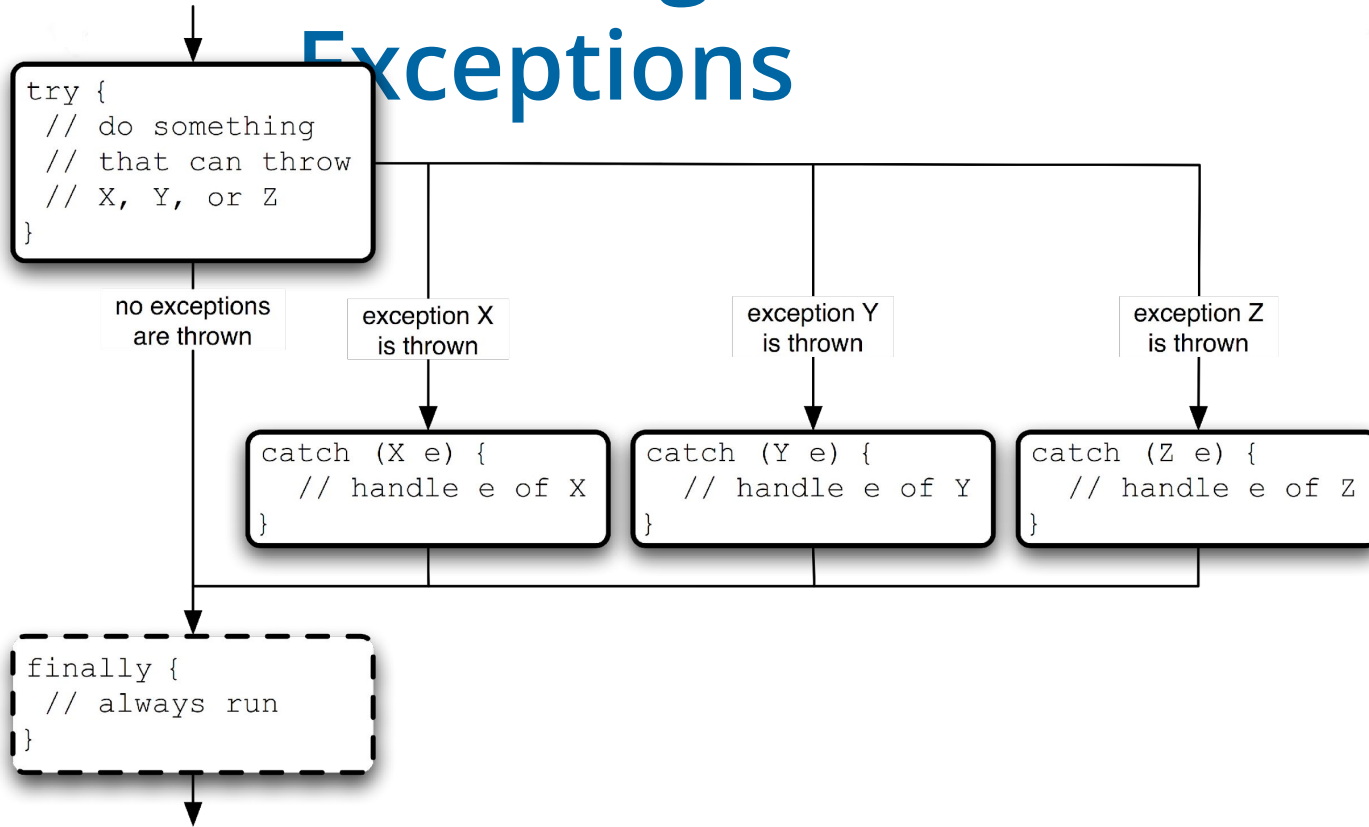
**StackOverflowError** – thrown by the JVM when a method calls itself too many times.

**NoClassDefFoundError** – thrown by the JVM when a class that the code uses is available at compile time but not runtime.





# Handling Exceptions





BUSINESS DEVELOPMENT  
GROUP

# Throwing Exceptions

A method that generates an unhandled exception is said to *throw* an exception:

- It generates an exception to signal an exceptional condition
- A method it calls throws an exception



# Contact US

Phone: +374 55 201 209

E-mail: [info@bdg.am](mailto:info@bdg.am)

Address: Hr. Kochar 4

Web: [www.bdg.am](http://www.bdg.am)

 <https://www.facebook.com/bdg.trainings/>

 [bdg.trainings](https://www.instagram.com/bdg.trainings/)

 <https://www.linkedin.com/in/bdg-trainings/>