THE UNIVERSITY OF MELBOURNE

COMP90041 Programming and Software Development

Tutorial 10 Exception

Chuang(Frank) Wang

THE UNIVERSITY OF MELBOURNE

Overview

1. Exception
2. Error
3. Exercise

THE UNIVERSITY OF
MELBOURNE

1. Exception

Exception

```
/Library/Java/JavaVirtualMachines/jdk-10.jdk/Contents/Home/bin/java -Dvisualv
  Documents/com~apple~CloudDocs/Master_Of_Software/2019_Sem2/COMP90041_Java/c
  Documents/com~apple~CloudDocs/Master_Of_Software/2019_Sem2/COMP90041_Java/c
Exception in thread "main" java.lang.NullPointerException
    at tute10.ExceptionDemo.nullpointerException(ExceptionDemo.java:17)
    at tute10.ExceptionDemo.main(ExceptionDemo.java:13)

Process finished with exit code 1
```

Examples

```
public static void example1(){
    int numerator = 10;
    int denominator = 0;
    System.out.println(numerator / denominator);
}
```

```
public static void example1(){
    String fruits[] = {"apple", "banana", "watermelon"};
    System.out.println(fruits[3]);
}
```

try-throw-catch trio (handle exceptions)

```
try {
      /*
      Some_Code_That_May_Throw_An_Exception
      */

} catch (Exception_Name e){
     /*
       Code to be performed if an exception of the
       named exception class is thrown in the try block.
     */
}
```

Definition

- ✓ Exceptions are events that occur during the execution of programs that disrupt the normal flow of instructions (e.g. divide by zero, array access out of bound, etc.)

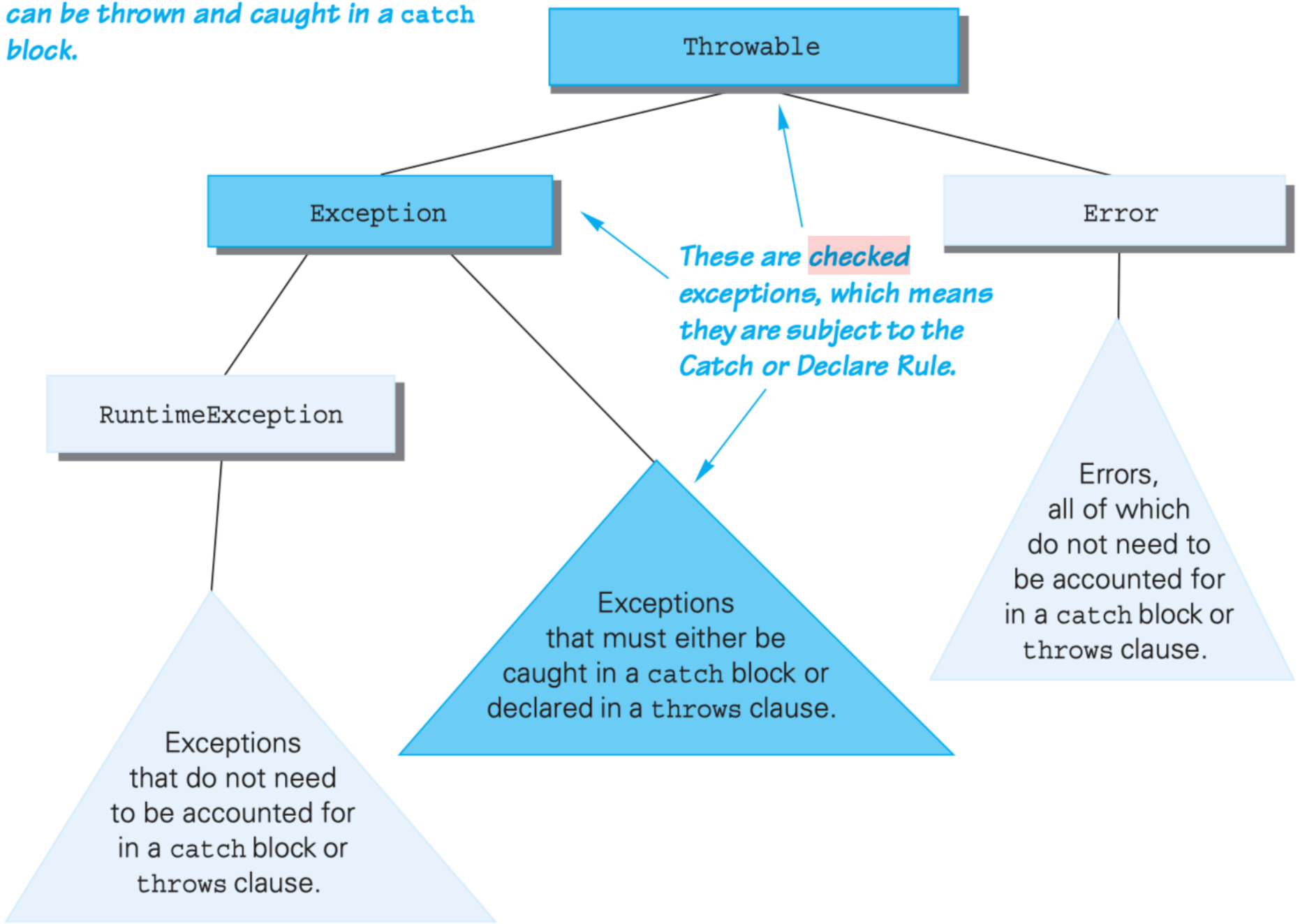
- ✓ **Checked Exception**

  Exceptions that are subject to the check and catch are called checked exceptions because the compiler checks to see if they are accounted for with a catch block or throws clause. e.g. SQLException

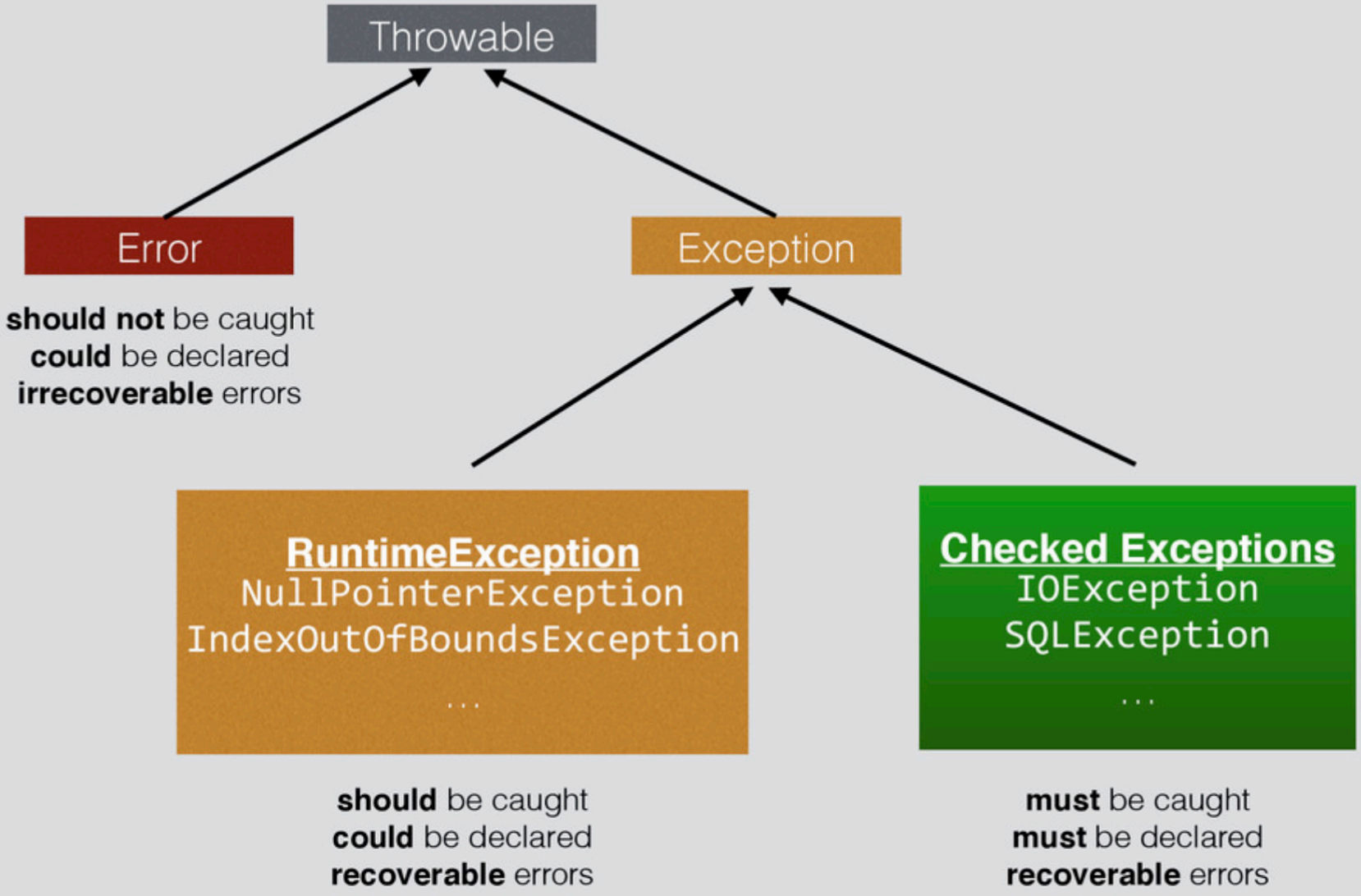
- ✓ **Unchecked exceptions**

  Exceptions that are not subject to the check and catch are called unchecked exceptions.

All descendents of the class `Throwable` *can be thrown and caught in a* `catch` *block.*



The Java Exception Hierarchy



Referenced from https://speakerdeck.com/gousiosg/exception-handling-bug-hazards-on-android

Why Use Exceptions?

Benefits

- ✓ Exceptions separate error handling code from regular code.
- ✓ Exceptions propagate errors up the call stack.
- ✓ Exception classes group and differentiate error types.
- ✓ Exceptions standardize error handling.

Create the custom exception class

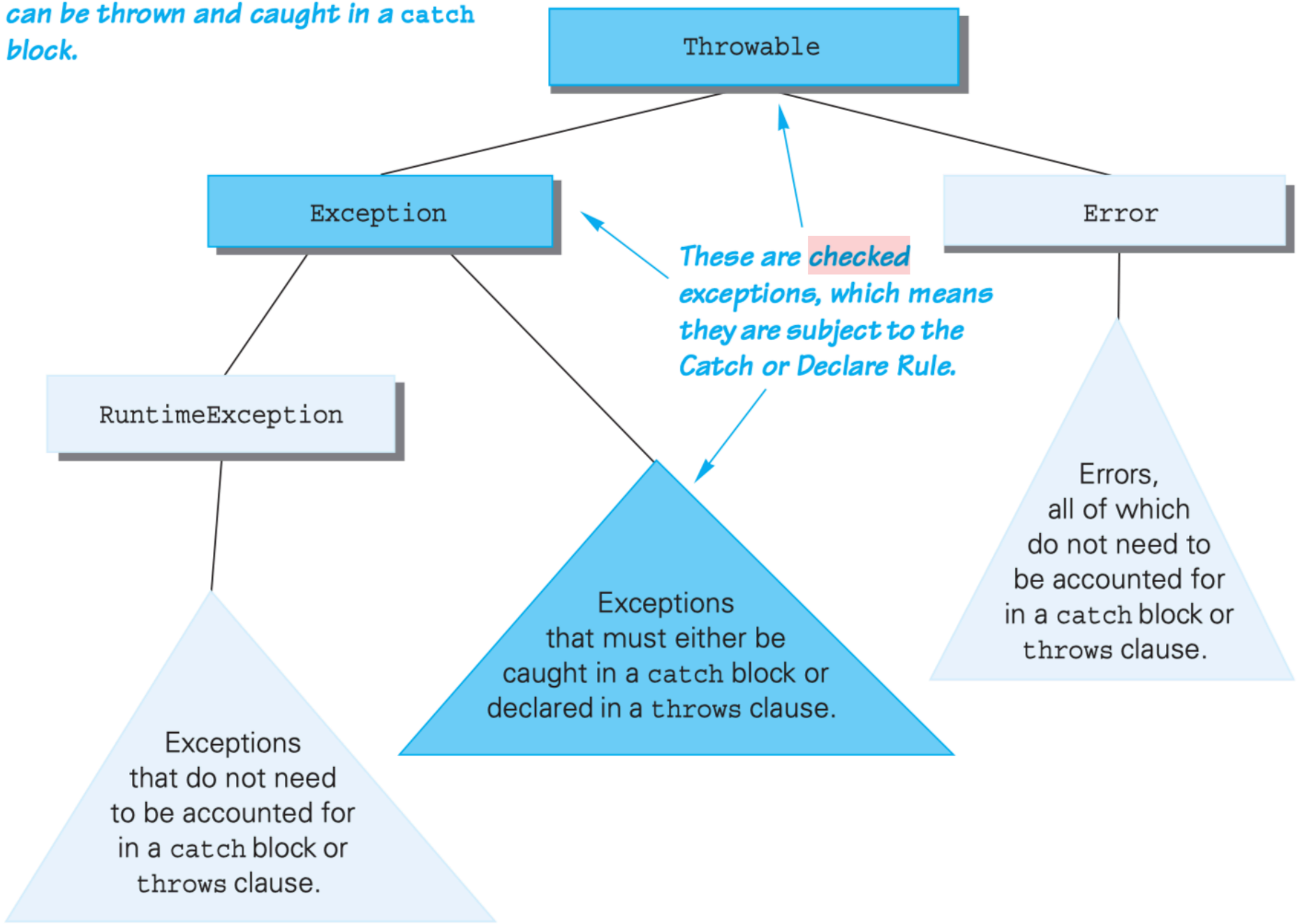
```
public class LoginFailException extends Exception {
    public LoginFailException(String msg) {
        super(msg);
    }
}
```

Demo

THE UNIVERSITY OF
MELBOURNE

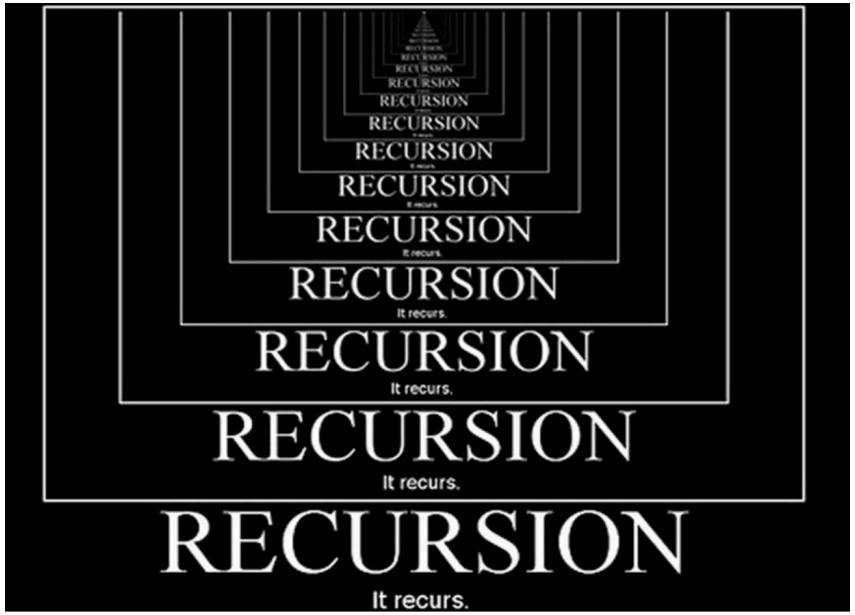
2. Errors

All descendents of the class `Throwable` *can be thrown and caught in a* `catch` *block.*



Errors

- ✓ Errors are thrown by JVM and should not be handled or declared.
- ✓ Errors indicate serious problems that a reasonable application should not try to catch. Most such errors are abnormal conditions.
- ✓ It is impossible to recover from errors.
- ✓ One example is StackOverflowError



Demo

THE UNIVERSITY OF
MELBOURNE

3. Exercise

Tutorial Exercise

1. Write a complete Java program that prompts the user for, and uses `nextInt()` to read in, two nonnegative integer numbers. Use a `try/catch` block to handle the `InputMismatchException` that is thrown if the user types in something other than an integer. In this case, print a suitable error message and exit the program.

Homework Exercise

2. Define an exception class called `NegativeNumberException`. The class should have a constructor with no parameters. If an exception is thrown with this zero-argument constructor, `getMessage` should return "Negative Number Not Allowed!". The class should also have a constructor with a single parameter of type String. If an exception is thrown with this constructor, then `getMessage` returns the value that was used as an argument to the constructor.

3. Revise the program in Exercise 1 above to use your new `NegativeNumberException`. Write a (static) method to print a prompt and read in a non-negative number. It should throw a `NegativeNumberException` if a negative number is entered. Then modify your `main` to catch `NegativeNumberException` as well as `InputMismatchException` and print a suitable error message in each case.

THE UNIVERSITY OF MELBOURNE

Thank you