Java Programming I

Session 8

Exceptions

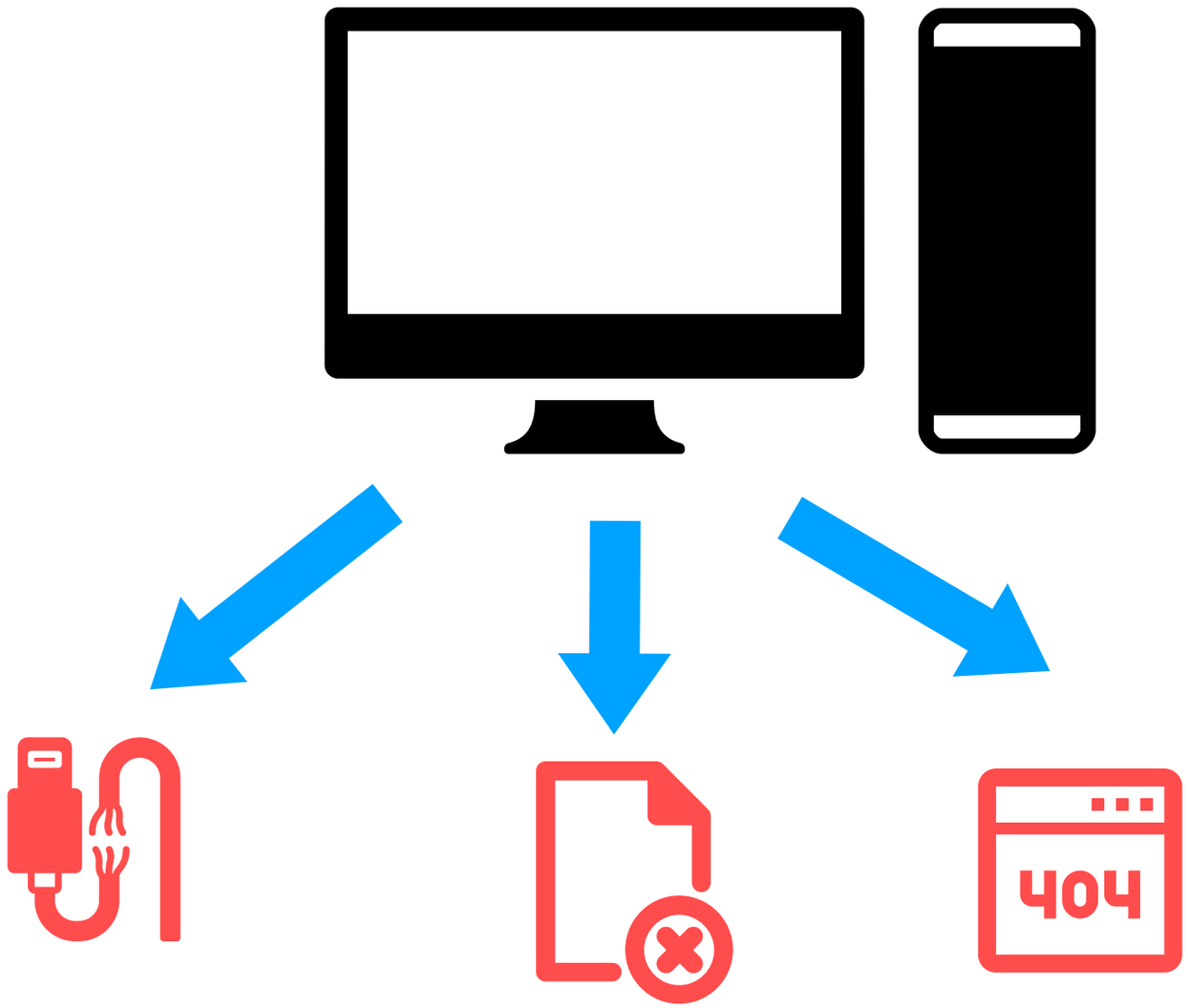
Juan Carlos Moreno - UCLA Ex

Agenda

- **Exception hierarchy**
- **try / catch**
- **Understanding of throwable**
- **finally**
- **Create your own throwable**
- **throws**

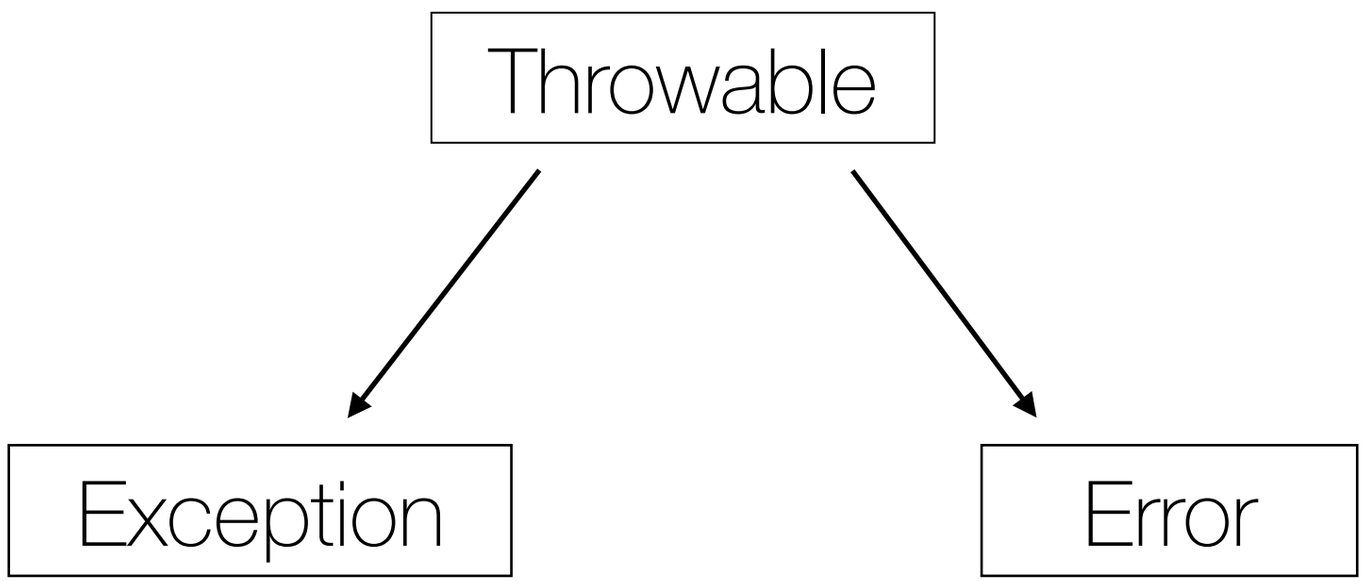
Exception

stuff happens



Exception

The hierarchy of exceptions



Program related

Unrelated to your code

Exception

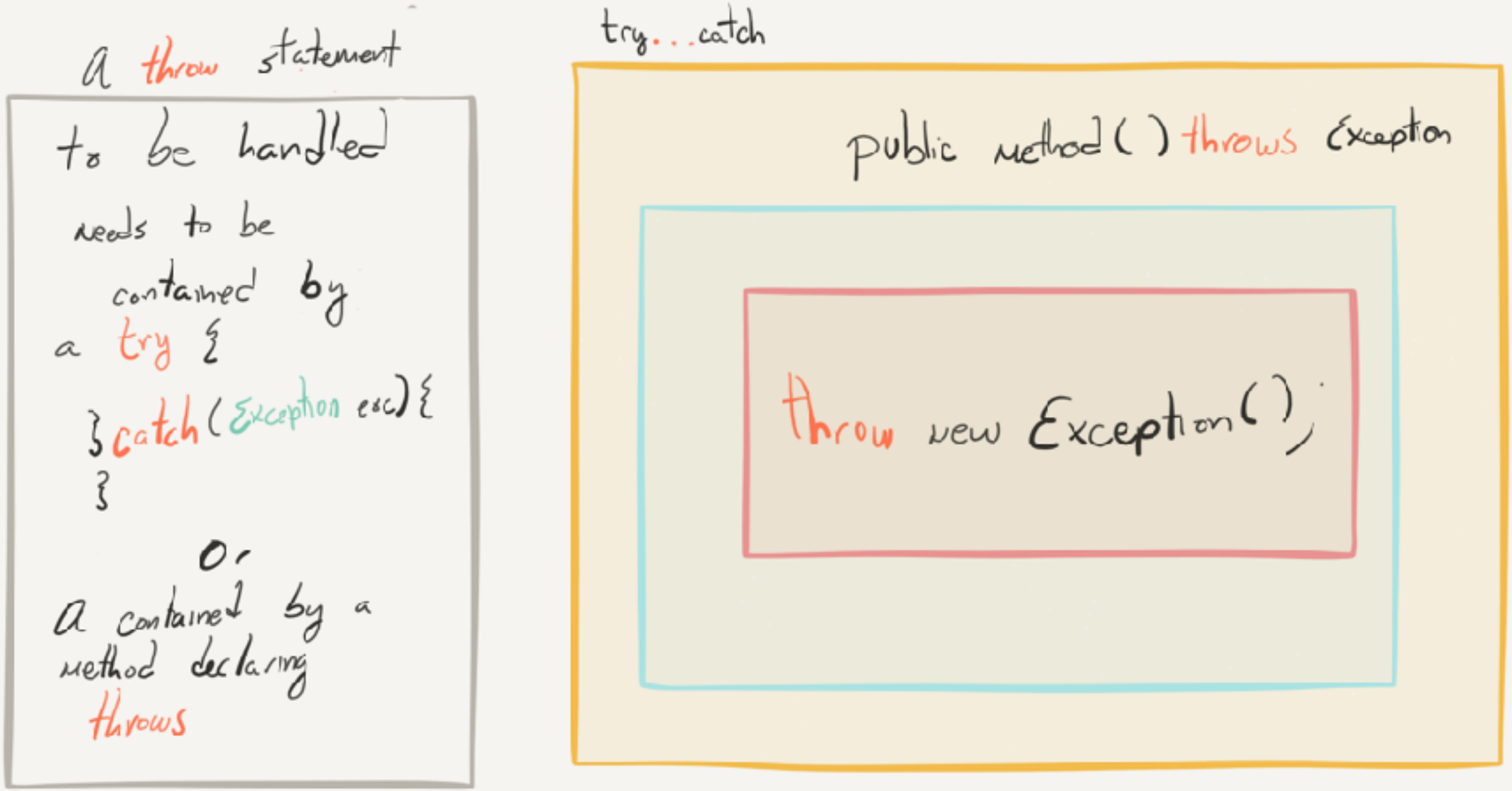
Unhandled exceptions

```
public class ExceptionExample {

    public static void main(String args[]){
        int arrays[] = {4,3,2,1};

        for (int i=0; i<5; i++)
        {
            System.out.println("Int is:" + arrays[i]);
        }

        System.out.println("Program has ended");

    }
}
```

```
Int is:4
objc[2394]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/bin/
java (0x10ceea4c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/jre/lib/libinstrument.dylib
(0x10cfb14e0). One of the two will be used. Which one is undefined.
Int is:3
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 4
Int is:2
     at ExceptionExample.main(ExceptionExample.java:8)
Int is:1

Process finished with exit code 1
```

Handling

Unhandled exceptions



Custom Throwable

DIY

```
import java.util.Scanner;
import java.io.FileReader;

public class ExceptionExample extends Throwable {


    public static void main(String args[]){
        Scanner scanner = new Scanner(System.in);

        try{
            ExceptionExample myerror = new ExceptionExample();
            throw myerror;
        } catch (ExceptionExample e) {
            System.err.println("Error " + e + "was thrown");
        }
    }
}
```

throwable

the root of all exceptions

Return Type	Method	Description
Throwable	fillInStackTrace()	throwable object that can be rethrown
String	getLocalizedMessage()	localized description of the exception
String	getMessage()	description of the exception
void	printStackTrace()	Displays the stack trace
String	toString()	Description of the exception

Common exceptions

A small list

Exception	Meaning
ArithmeticException	Arithmetic error (divide by zero)
ArrayIndexOutOfBoundsException	Access non-existing index of an array
ArrayStoreException	Assignment to an incompatible type
ClassCastException	Invalid cast
IllegalArgumentException	Argument to invoke method is invalid
NegativeArraySizeException	Array created with negative size
NullPointerException	Null reference
NumberFormatException	Invalid conversion from string to int
StringIndexOutOfBoundsException	Indexing non-existing element of String

finally

regardless do this

```
import java.util.Scanner;
import java.io.FileReader;

public class ExceptionExample {

    public static void main(String args[]){
        Scanner scanner = new Scanner(System.in);

        try{
            System.out.println("First number:");
            float first = scanner.nextFloat();
            System.out.println("Divided by:");
            float second = scanner.nextFloat();

            float result = first/second;
            System.out.println("The result is:" +  result);
        } catch (Exception e) {
            System.err.println("Error " + e.getMessage());
        } finally {
            System.err.println("Thanks for participating");
        }
    }
}
```

Chained Exceptions

Catching different exceptions

```
public class MyBank {

    public static void main(String argvs[]){
        ATM corner_atm = new ATM();
        try{
            corner_atm.widthdraw(1234, 800);
        }catch (InsufficientFundsException exc){
            System.out.println("No money, mo' problems");
        } catch(IncorrectPinException exc){
            System.out.println("Your pin is wrong!!");
        } catch (Exception exc){
            System.out.println("Yeah I don't what happened");
        } finally {
            System.out.println("Hey, grab your card!");
        }
    }
}
```

Throwing Exceptions

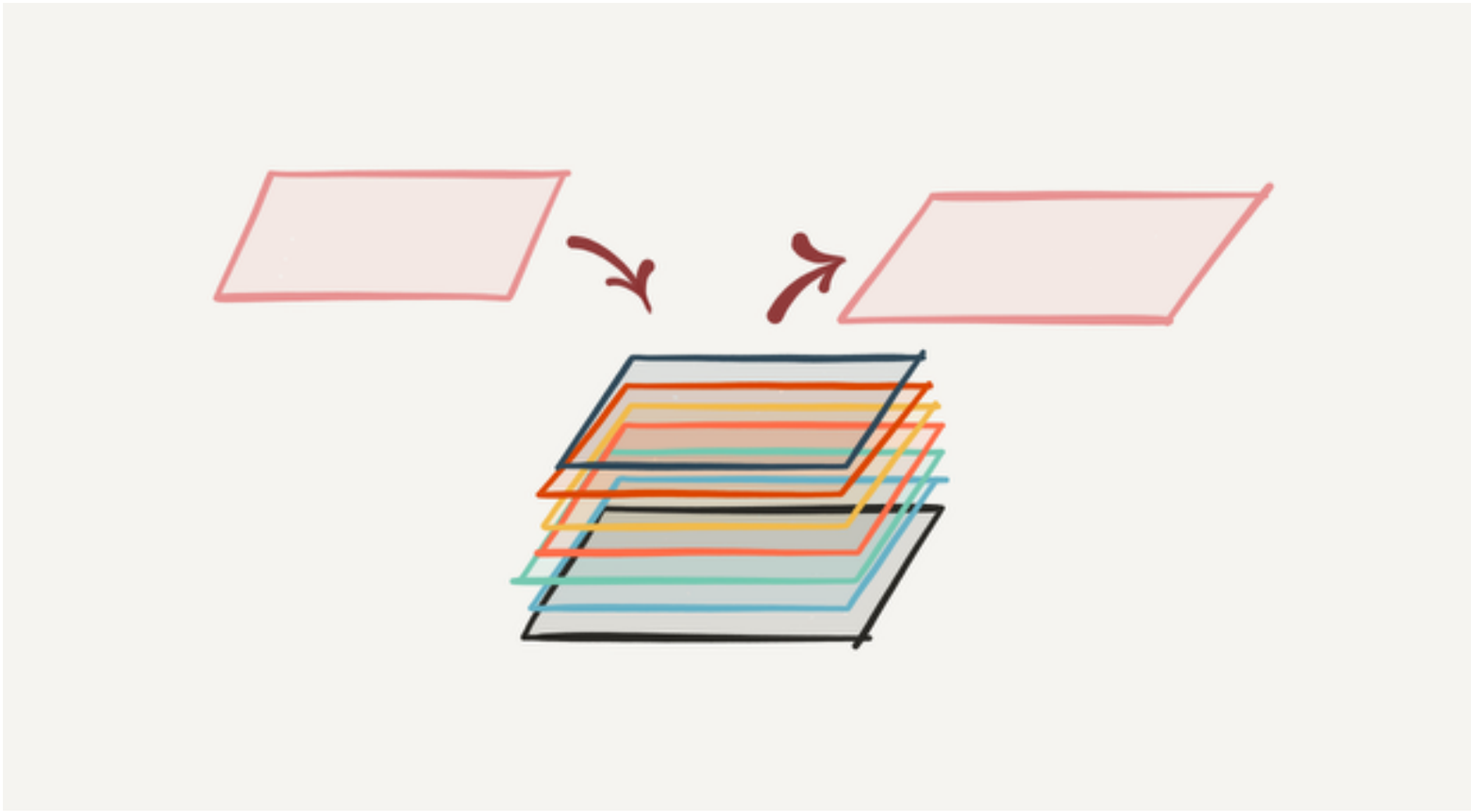
```
public class ATM {
    float balance = 1000;

    public float widthdraw(int pin, float amount) throws
Exception{
        if (pin != 1234) // Lets assume the DB returned this
        {
            throw new IncorrectPinException();
        }

        if (amount> balance)
        {
            throw new InsufficientFundsException();
        }
        return amount;
    }
}
```

Custom Throwable

```
public class IncorrectPinException extends Exception
{
}



public class InsufficientFundsException extends Exception
{
}
```

Stack

Papers, dishes, coins



- LIFO - Last in, First out
- push - puts an item on top
- pop - pop, returns and removes the item on the top

DIYStack

Build your own stack

```
public class DIYStackException extends Throwable{

}

public class DIYStackNode {

}

public class DIYStack {

    public static void main(String args[]){
        DIYStack stack = DIYStack();
        // Do work here
    }
}
```

PasswordCheck

Verify that 2 passwords match

```
function passCheck:
    pass1 = <user prompt>
    pass2 = <user prompt>
    if pass1 != pass2:
        throw exception

function main:
    for each passCheck fail
        print “Try again”
        if 3 attempts fail:
            print “Security warning”
            exit
```