

CS1130 14 February 2014
 Exceptions in Java. Read chapter 10.

HUMOR FOR LEXOPHILES (LOVERS OF WORDS):

Police were called to a day care; a three-year-old was resisting a rest.
 Did you hear about the guy whose whole left side was cut off?
 He's all right now.

The butcher backed into the meat grinder and got a little behind in his work.

When fish are in schools they sometimes take debate.

A thief fell and broke his leg in wet cement. He became a hardened criminal.

Thieves who steal corn from a garden could be charged with stalking.

When the smog lifts in Los Angeles, U.C.L.A.

Exceptional circumstances

```
/** = the decimal number represented by s. */
int parseInt(String s) { ... }
...but what if s is "bubble gum"?
```

```
/** = the decimal number represented by s, or -1 if s
    does not contain a decimal number. */
...but what if s is "-1"?
```

```
/** = the decimal number represented by s
    Precondition: s contains a decimal number. */
...but what if s might not, sometimes?
```

Somehow, we have to be able to deal with the unexpected case.

Dealing with exceptional circumstances

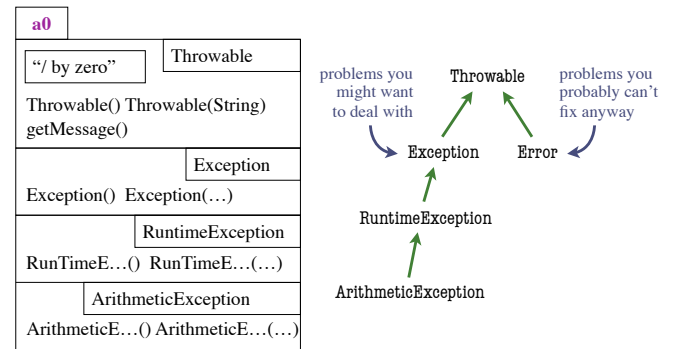
```
// String s is supposed to contain an integer.
// Store that integer in variable b.
b= Integer.parseInt(s);
```

```
/** Parse s as a signed decimal integer and return
    the integer. If s does not contain a signed decimal
    integer, throw a NumberFormatException. */
public static int parseInt(String s)
```

parseInt, when it find an error, does not know what caused the error and hence cannot do anything intelligent about it. So it "throws the exception" to the calling method. The normal execution sequence stops!

Exceptions in Java

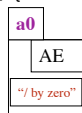
Exceptions are represented by instances of class Throwable.
 Making exceptions instances of classes lets them be organized in a hierarchy.



```
Class -----> 02 /** Illustrate exception handling */
                03 public class Ex {
                04     public static void first() {
                05         second();
                06     }
                07
                08     public static void second() {
                09         third();
                10     }
                11
                12     public static void third() {
                13         int x= 5 / 0;
                14     }
                15 }
```

Call:
 Ex.first();

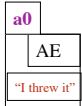
Output:
 ArithmeticException: / by zero
 at Ex.third(Ex.java:13)
 at Ex.second(Ex.java:9)
 at Ex.first(Ex.java:5)



```
Class -----> 02 /** Illustrate exception handling */
                03 public class Ex {
                04     public static void first() {
                05         second();
                06     }
                07
                08     public static void second() {
                09         third();
                10     }
                11
                12     public static void third() {
                13         throw new ArithmeticException ("I threw it");
                14     }
                15 }
```

Call:
 Ex.first();

Output:
 ArithmeticException: I threw it
 at Ex.third(Ex.java:13)
 at Ex.second(Ex.java:9)
 at Ex.first(Ex.java:5)



Example of a throwable subclass

```

/** An instance is an exception */
public class OurException extends Exception {

    /** Constructor: an instance with message m */
    public OurException(String m) {
        super(m);
    }

    /** Constructor: an instance with no message */
    public OurException() {
        super();
    }
}

```

```

Class -----> 02 /** Illustrate exception handling */
                03 public class Ex {
                04     public static void first() throws OurException {
                05         second();
                06     } // "This method sometimes throws OurException"
                07
                08     public static void second() throws OurException {
                09         third();
                10     }
                11
                12     public static void third() throws OurException {
                13         throw new OurException("Whoa!");
                14     }
                15 }

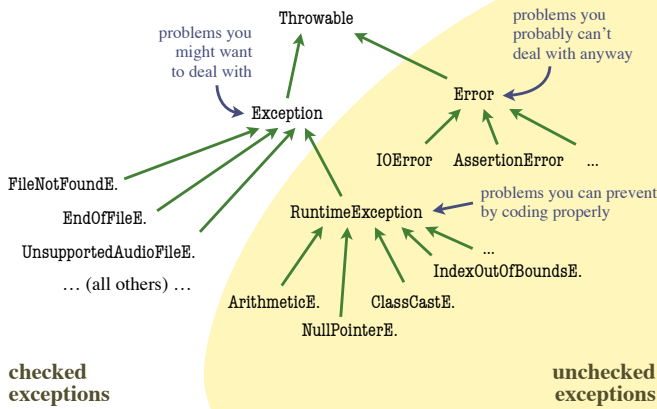
```

Call:
Ex.first();

OurException: Whoa!
at Ex.third(Ex.java:13)
at Ex.second(Ex.java:9)
at Ex.first(Ex.java:5)

throws clauses are required because OurException, unlike ArithmeticException, is a "checked exception".

Exception Hierarchy



Catching a thrown exception

```

public class Ex1 {
    public static void first() {
        try {
            second();
        } catch (MyException ae) {
            System.out.println("Caught MyException: " + ae);
        }
        System.out.println("Procedure first is done.");
    }
    public static void second() throws MyException {
        third();
    }
    public static void third() throws MyException {
        throw new MyException("yours");
    }
}

```

Execute the try-block. If it finishes without throwing anything, fine. If it throws a MyException object, catch it (execute the catch block); else throw it out further.

```

/** Input line supposed to contain one int, maybe whitespace on either side. Read line, return the int. If line doesn't contain int, keep asking until it does. */
public static int readLineInt() {
    String input= readString().trim();

    // inv: input contains last input line read; previous
    // lines did not contain a recognizable integer.

    while (true) {
        try {
            return Integer.valueOf(input).intValue();
        } catch (NumberFormatException e) {
            System.out.println("Input not int. Must be an int like");
            System.out.println("43 or -20. Try again: enter an int.");
            input= readString().trim();
        }
    }
}

```

Wed, 22 Feb, in Holl B14. At 2:30, 4:00, 7:00 (1.5 hour test but easily doable in 1 hour)

Take it either time, but some of you in the 2:30 section please take it at a later time to avoid overcrowding.

You do not HAVE to take it on Wednesday. Please take it ONLY when you believe you can pass. Don't waste our time and yours. We will give the test at the same time on Friday and then the next week, Mon-Wed-Fri.

Take the test several times until you pass. **Don't waste our time by taking it when you are not ready!**

The course website has (or will have) a list of things you are expected to know and sample questions.