Exceptions

- Exceptions are your friends. They allow you to control the result of unexpected things happening in your program, often things caused by stupid users that you are trying to persuade to give you money so you can’t tell them they are being stupid….

Example1

This will crash if the user enters 0.

```
while (!done) {
        System.out.print( "Enter a number ");
        int x = input.nextInt();
        if (x < 0)
                done = true;
        else
                System.out.printf(“%d \n",  100/x);
}
```

Example 2
This catches the exception

```
•   while (!done) {
            System.out.print( "Enter a number ");
            int x = input.nextInt();
            if (x < 0)
                    done = true;
            else  try {
                    System.out.printf("%d \n", 100/x);
            }
            catch(ArithmeticException e) {
                    System.out.println( "oops; can't divide by zero.");
            }
}
```

When you call a method that throws a “checked” exception you must either catch it or throw it on

```
public FilePrinter(String fName) throws
                         FileNotFoundException {

      Scanner fileIn = new Scanner(new
                            FileReader(fName));
```

Try-blocks can have “Finally” clauses

```
try {
        System.out.println( 100/x);
        return;
}
catch (ArithmeticException e) {
        System.out.println( "oops");
}
finally {
        System.out.println( "finally");
}
```

The finally clause is always executed – if there is no exception, if there is a caught exception, or if there is an uncaught exception. Even a return statement in the try-clause does not prevent the finally clause from being executed.

Common Exceptions

Runtime:

- IndexOutOfBoundsException
- NumberFormatException
- NullPointerException
- ArithmeticException

Checked:

- EOFException
- FileNotFoundException
- IOException

See Weiss p. 50 or a huge list of exceptions at
http://mindprod.com/jgloss/exception.html