COMP202
Instructor: Ladan Mahabadi
April 9, 2008

Throwing exceptions

A method will throw an exception when the method that detects an error is unable to deal with it. There is no guarantee that an error will be handled but utilizing throws will enable you as a programmer to separate the portion of a program that handles exceptions from the portion that generates it.

A throw statement specifies an (exception) object to be thrown. This object needs to be derived from class Throwable (which includes classes Exception and Error as subclasses).

Errors: serious system problems that generally should not be caught
Exceptions: problems that are caused by programs; (should) deal with these

Exception Handlers:

- contained in catch blocks
- in the catch brackets you need to indicate the type of exception that's being caught here and a parameter name
- a catch can only have a single argument (can't use comma seperated arguments)
- You can not handle the same type of error in two different catch blocks associated with a particular try block
- Placing catch(Exception e) before other catch blocks prevents other (specific) types of exceptions from being executed; catch(Exception e) should always be placed last in the list of exception handlers following a try block

Rethrowing an Exception

A catch handler that catches an exception may determine that it can not handle that error well and it may allow another catch handler to deal with the problem. In these cases, the handler can rethrow the exception to the next enclosing try block by (even if it has processed it a bit) :

```
throw e;
```