Not Business As Usual

Jonathan Lung – CSC207H: Software Design
What Are They Good For?

- Exceptions report exceptional conditions: unusual, strange, disturbing.
- These conditions deserve exceptional treatment: not the usual go-to-the- next-step, plod-onwards approach.
- They break the normal flow of execution to signal something exceptional.
Before Exceptions...

- ...we used special return values for functions to signify an error.
- ...we used flags to signal a condition.
The Big Picture

- To signal an special condition, a piece of code *throws* an exception.
- To handle a special condition, a piece of code *tries* to execute a block of code and *catches* any exceptions thrown inside that block.
- We put clean up code that runs *finally*, regardless of whether an exception was encountered or not.
Two Kinds of Exceptions

- **Runtime exceptions** “can be thrown during the normal execution of the Java Virtual Machine” \(^1\). E.g., `IndexOutOfBoundsException`, `NullPointerException`, and `UnsupportedOperationException`.
- Every other kind of exception (**checked exceptions**).

\(^1\)http://java.sun.com/j2se/1.4.2/docs/api/java/lang/RuntimeException.html
Can’t Catch This (or at least generally shouldn’t)

- catch (Throwable thr) { ...} – this is too broad, unless you just want to play ostrich.
- catch (Exception e) { ...} – this is too broad, unless you just want to play ostrich.
- While not an exception, we can also catch Errors, but an Error “indicates serious problems that a reasonable application should not try to catch.”\(^2\)

\(^2\)http://java.sun.com/j2se/1.4.2/docs/api/java/lang/Error.html
Declaring “throws”

Consider `public void x() throws Y, Z {...}`.

- Any uncaught exceptions that belong to the classes (or subclasses of) Y or Z will be propagated (“re-thrown”) automatically.
- Any uncaught checked exceptions that are not instances of Y or Z (or their subclasses) will trigger a compiler error.
- `x()` reserves the right to throw a Y or a Z, so anything that calls `x()` needs to be prepared.
Alternatives to Exceptions

- Exceptions are relatively heavyweight solutions in terms of memory usage.
- What if we may not have memory to create a new exception object?
- One solution employed in Symbian OS is the use of LEAVEs and TRAPs.