

Checked vs Unchecked Exceptions

- *checked exceptions* are the ones the compiler forces you to deal with
 - e.g. `IOException`
 - must either enclose with `try..catch` or declare `throws`
 - meant for conditions that a caller could reasonably anticipate and recover from – a file not existing, a network connection failing
- *unchecked exceptions* (`RuntimeException` and its subclasses) result in a crash if not caught
 - e.g. `IllegalArgumentException`, `NullPointerException`
 - meant for conditions that indicate a programming error where recovery isn't possible – solution is to fix the bug

Choosing and Using Exceptions

- key factor in choosing checked vs unchecked: *is this a condition the caller could reasonably anticipate and recover from, or a sign that something upstream already went wrong?*
- don't catch an exception just to silence it
 - an empty catch block hides real problems
- exception messages should say what happened and, where possible, what value caused it
 - e.g. "n must be between 1 and the column size, got "+n
- don't use exceptions for ordinary control flow
 - reserve them for actual exceptional conditions