# Exception Types

# Fatal

**Definition:** Fatal exceptions are *not your fault*, and you *cannot sensibly clean up from them*.

**Examples:** Out of memory, thread aborted.

**Resolution:** Don’t catch; let them crash the program.

**Design:** Don’t ever throw fatal exceptions directly.

# Boneheaded

**Definition:** Boneheaded exceptions are *violations of the API*, and are *bugs in your code*.

**Examples:** Argument is null, index out of range.

**Resolution:** Don’t catch; fix them in the code.

**Design:** Use code contracts for boneheaded exceptions; do not document the specific exception type.

# Vexing

**Definition:** Vexing exceptions are due to *bad design decisions*, thrown in *non-exceptional situations*.

**Examples:** Parsing errors.

**Resolution:** Avoid calling vexing functions; if not possible, catch the vexing exception.

**Design:** Don’t ever throw vexing exceptions.

# Exogenous

**Definition:** Exogenous exceptions are from *unpredictable, external influences*.

**Examples:** File not found, resource already in use.

**Resolution:** Always catch and handle.

**Design:** Throw exogenous exceptions as necessary; document the specific exception type.