Test-Driven Development

Problem: Buggy Software

Errors ⇒ Implementation doesn’t satisfy specification

OR specification is wrong or incomplete/inconsistent

Spec. (ADT) ?? Impl. (code)
Test-Driven Development:

1. write a test, based on spec.
2. write code to make the test work
3. incomplete? (case missing)

Black-box testing →
all based on spec — what's visible from the outside.
Adversarial roles — Devil’s advocate

- Write a test — try to expose a problem
- If test fails, modify the code in the simplest way to make the test succeed

⇒ Seems silly at first.
Exception Handling: The basics

```java
try {
    // something in here may throw an exception
}
catch (Exception e) {
    // handle the exception
}
```
try {
    open some file
} catch (FileNotFoundException fnfe) {
    open file choose dialog
    then try again
}
Glass Box Testing:
  Look inside the code to construct tests that cover all branches.

Black Box:
  Only deal w/ the specification

Adv: General \implies
  if you change the impl or make a new one,
  use the same tests