

Common Strategies for Software Testing

1. Test-driven development:

- write tests first, then
- write code to pass tests.
- Don't write code unless a test demands it.

2. Clear/white-box testing:

- write code based on specification, and
- write tests based on code.

Both are useful, and both can be abused.

5

Extreme Case of a Real Pitfall: Test-Driven Development

```
switch (test_number) {
    case 0: printf ("Answer 0");
            break;
    case 1: printf ("Answer 1");
            break;
    // and so forth...
}
```

Designed to pass tests, not to do something.

6

Real Pitfall: Clear Box Testing Can Miss the Point

Tests based only on code, so

- if developer forgets some functionality,
- there are no tests for it
- (it's not in the code!).

To be fair, clear box testing is

- usually used for **unit testing** (low-level component testing);
- **integration testing** (of the whole program) is supposed to handle missed cases.

7

Use Both Approaches in Practice

Write tests

- **based on intended behavior**
- representing **corner cases** when possible.

Write code and use tests to debug.

Write more tests

- **based on code**
- to ensure that all code works as intended
- and is necessary.

8