

Chapter 11 - Testing

- Expect abuse and plan for it
- Big picture understanding
- Knowing how the pieces fit together
- Organization
- Practicality
- Testing a real world object
 - Who will use it and why?
 - What are the use cases?
 - What are the bounds of use?
 - What are the stress/failure conditions?
 - How would you perform testing?
- Testing a piece of software
 - Black box testing vs white box testing?
 - Who will use it and why?
 - What are the use cases?
 - What are the bounds of use?
 - What are the stress conditions/failure conditions?
 - How would you perform testing?
- Testing a function
 - Define the test cases
 - normal case
 - the extremes
 - nulls and illegal input
 - strange input
 - Define the expected result
 - Write test code
- Troubleshooting questions
 - Understand the scenario
 - Break down the problem
 - Create specific, manageable tests

// **Mistake** → find the mistake in the following code

```
unsigned int i;
for (i = 100; i >= 0; --i) {
    printf("%d\n", i);
}
```

// Random crashes → some possible causes

- random variable
- uninitialized variable → take on an arbitrary value
- memory leak
- external dependencies

// Chess Test → extreme case validation and general case testing

// No Test Tools → load test a web page

- Response time
- Throughput
- Resource utilization
- Maximum load the system can bear

// Test a pen

// Test an ATM