



Testing & Debugging

Yannis Smaragdakis, Alex Orso, CS 1322



What is Testing?

Testing == To execute a program with a sample of the input data

- Goal: Improve software quality by finding errors
 - “A Test is successful if the program fails”
- Optimistic approach:
 - The program under test is exercised with a (very small) subset of all the possible input data
 - We assume that the behavior with any other input is consistent with the behavior shown for the selected subset of input data

Yannis Smaragdakis, Alex Orso, CS 1322

Testing Techniques



There exists a number of techniques

- Different processes
- Different artifacts
- Different approaches

There are no perfect techniques

- Testing is a best-effort activity

Yannis Smaragdakis, Alex Orso, CS 1322

Exhaustive Testing?



- How long would it take (approximately) to test exhaustively the following program?

```
int sum(int a, int b) {  
    return a + b;  
}
```

- $2^{32} \times 2^{32} = 2^{64} \approx 10^{19}$ tests
- Assume 1 test per nanosecond (10^9 tests/second)
- we get 10^{10} seconds...
- About 600 years!
- In general, exhaustive testing is not an option

Yannis Smaragdakis, Alex Orso, CS 1322

Failure, Fault, Error



Failure

Observable incorrect behavior of a program. Conceptually related to the behavior of the program, rather than its code

Fault (bug)

Related to the code. Necessary (not sufficient!) condition for the occurrence of a failure

Error

Cause of a fault. Usually a human error (conceptual, typo, etc.)

Yannis Smaragdakis, Alex Orso, CS 1322

Failure, Fault, Error: Example



```
1. int double(int param) {
2.     int result;
3.     result = param * param;
4.     return(result);
5. }
```

- A call to double(3) returns 9
- Result 9 represents a **failure**
- Such failure is due to the **fault** at line 3
- The **error** is a typo (hopefully)

Yannis Smaragdakis, Alex Orso, CS 1322

Oracle



An **oracle** predicts the expected results of a test and is used to assess whether a test is successful or not.

There are different kinds of oracles:

- Human (tedious, error prone)
- Automated (sometimes expensive)
- Approximated

Yannis Smaragdakis, Alex Orso, CS 1322

Granularity Levels



Unit testing: verification of the single classes

Integration testing: verification of the interactions among the different classes

System testing: testing of the whole program

Yannis Smaragdakis, Alex Orso, CS 1322

Debugging



Activity that takes place after revealing a failure

- Identify the fault(s) responsible for the failure
- Fix the problem

Yannis Smaragdakis, Alex Orso, CS 1322

Program Development Advice



- DON'T
 - write all first, test next
 - complain of vague errors. Be specific, look for minimal example
- DO
 - develop incrementally
 - what you have should always run and then you just add more functionality
 - develop unit tests
 - isolate errors / divide and conquer
 - dynamically: what's the difference between your program and working code?
 - logically: test small pieces with unit tests
- Programming = divide and conquer

Yannis Smaragdakis, Alex Orso, CS 1322