

Testing

- Important Consideration – design effective test cases
- Testing will NOT guarantee the absence of errors
- What subset of all possible test cases has the highest probability of detecting the most errors

Testing

- Definition – executing the program with the intent of finding errors
- Good Test – one that has a high probability of finding an error
- Successful Test – one that finds an error

Testing Principles

1. Necessary part of a test plan is **expected output**
2. A programmer should avoid testing his own program
3. A programming test should avoid testing its own programs
4. Inspect the results of each test case

Testing Principles

5. Test cases must be written for **invalid** and **unexpected** input
6. Examine a program to see if it does what it is not suppose to do
7. **Avoid** throw away test cases
8. Do not plan a testing effort under the assumption that no errors will be found

Testing Principles

9. The probability of the existence of more errors in a section of a program is directly preportional to the number of errors already found in that section
10. Testing is an extremely creative and challenging task

Who reads programs?

- Computers or Humans?
“Human Testing”
When – **after** the code is written
before the code is tested

Human Testing

1. The earlier an error is found the “cheaper” it is to fix
2. Programmers change when machine testing begins
50% chance of creating an error when fixing one.

Walkthroughs

- Also termed “code inspections”
- 3 people
 - Programmer
 - Designer
 - Outsider

Walkthroughs

Arrange the Walkthrough 24 hours in advance
Hand out the syntactically correct code 24 hours before
“Play Computer”
comments directed at the program
Find errors, don't **fix** them

Walkthroughs

- Time – 50 lines of code/30 mins