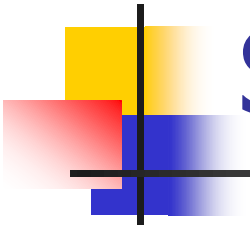
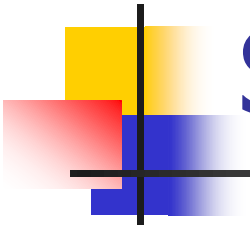


Software Testing

- A naïve manager may think a fresh graduate makes a good tester –

- Software testing is a specialized field that requires specific skills
- Many managers erroneously assume that a programmer can test software effectively without training or mentoring



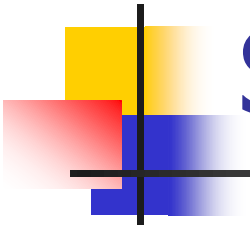
Software Testing

- Novice:
  - Knows where to end up
  - Does not know how to get there, and
- In case you didn’t know:
  - The product ships next Friday!!
  - The project document is incomplete

Software Testing

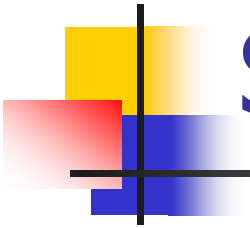
"Would you tell
me please,
which way I
ought to go
from here?"
"That depends
a good deal
on where you
want to get to,"
said the cat.
"I don't much
care where—,"
said Alice.
"Then it
doesn't matter
which way
you go," said
the cat.

Lewis Carroll



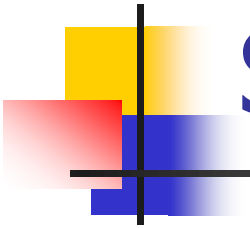
Software Testing

- A good software development environment provides:
  - Adequate staffing
  - Guidance
- Some testers, however,
  - work under less ideal conditions
  - test a new application under unreasonable deadlines



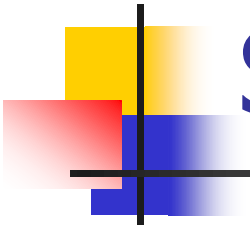
Software Testing

- Some organizations will have already decided the shipping date and want testers to confirm the product's suitability for use …
  - Not the best way to test, but
  - The tester may have no other choice
- To be effective, a software tester should be knowledgeable in two areas:
  - Software testing techniques
  - The application-under-test



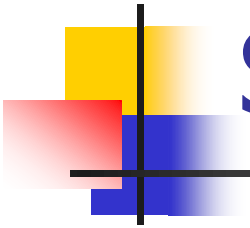
Software Testing

- A tester must invest time in learning about the application
- A tester with no experience must also learn:
  - Testing techniques
  - How to define suitable list of tests



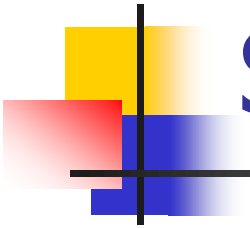
Software Testing

- To accommodate tight deadline:
  - Determine if application works under normal conditions
  - In later stages, try to break the system ..



Software Testing

- A mathematician, a physicist, and an engineer are told, “All odd numbers are prime.”
- The mathematician response:
- The physicist:
- The engineer:



Software Testing

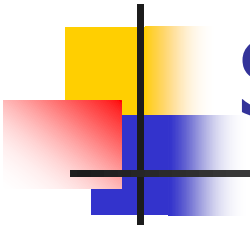
- Testing is like the above ‘joke’
  - A tester is given a false statement:
    - “The system works”

    and has the job of selecting, from an infinite number of possibilities, an input that contradicts the statement

  - Avoid both the physicist’s error:

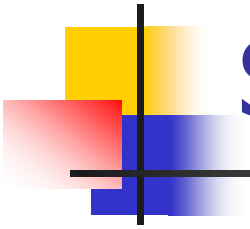
    and the engineer’s

  - You must be like the mathematician



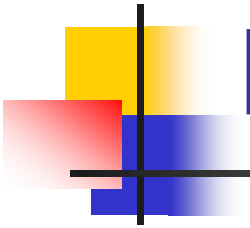
Software Testing - Terminology

- Error:
  - A mistake made by the developer; Maybe:
    - A typographical error
    - A misreading of the specification
    - A misunderstanding of what a subroutine does, etc
  - Mostly located in people's heads



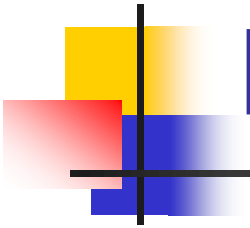
Software Testing - Terminology

- Fault:
  - An error may lead to one or more faults
  - Located in the text of the system
  - A fault is the difference between an incorrect program and a correct version
    - Example: We may describe a fault as



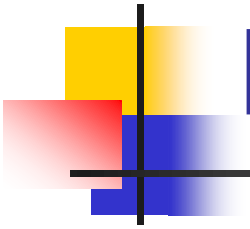
Fault Types

- Algorithmic
- Computation and precision
- Documentation



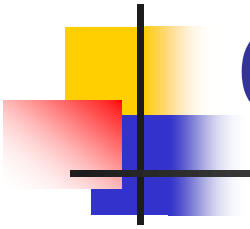
Fault Types

- Stress/Overload
- Capacity/Boundary
- Throughput/Performance
- Recovery



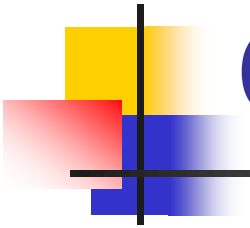
Defect Classification Scheme

- Orthogonal
  - Classified item belongs to exactly one category



Orthogonal Array Testing

- Even for small number of input values, listing all permutations of values generates massive number of tests
- Orthogonal array testing reduces the number of test cases



Orthogonal Array Testing

- Based on statistical technique borrowed from manufacturing
- Application:
  - Requires independent sets of states
  - The goal is to pair states

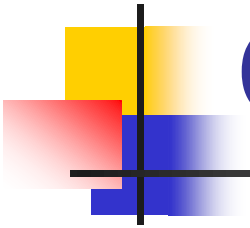
Orthogonal Array Testing

- Example:

  A bookstore processes the following information

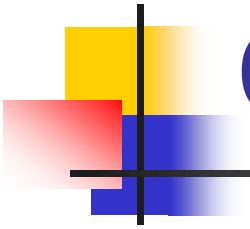
Book	**Purchase**	**Shipping**
in_stock	cash	overnight
special_order	check	economy
out_of_print	charge	ground
		pick-up

Classes and states in the bookstore example



Orthogonal Array Testing

- In OO application, the classes book, purchase, and shipping each has a finite number of possible states
- In a procedural application, there'd be three procedures each having arguments with a finite set of values



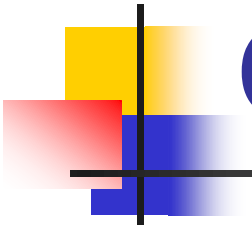
Orthogonal Array Testing

- Two classes have three states and one has four
- Selecting a state from each of two classes (pair-wise combination) only requires the 12 test cases shown (Next slide)

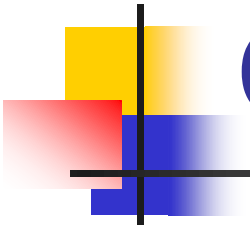
Applying Orthogonal Arrays

Test case	Book	Purchase	Shipping
1	in-stock	cash	overnight
2	in-stock	check	economy
3	in-stock	charge	ground
4	in-stock	cash	pick-up
5	special_order	check	overnight
6	special_order	charge	economy
7	special_order	cash	ground
8	special_order	check	pick-up
9	out_of_print	charge	overnight
10	out_of_print	cash	economy
11	out_of_print	check	ground
12	out_of_print	charge	pick-up

Orthogonal Array Testing



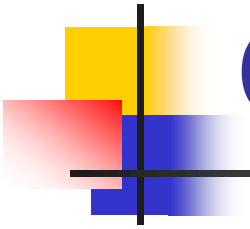
- Take any two (class) columns, say book and shipping, the array shows every possible combination of these two classes
- Shipping has more states than the other two classes; inspecting the pairings of the book and purchase columns shows every possible combination with some duplication



Orthogonal Array Testing

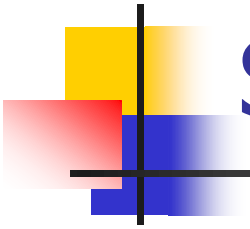
- What does Test case 1 say?
  - How would this test case be implemented in
    a) an OO environment?
    b) a procedural language?

NOTE: *Submit your written response next class*



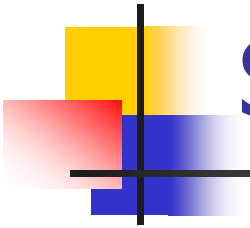
Orthogonal Array Testing

- Are there any combinations which seem improbable? if so,
  - Should the tester view these as non-feasible and remove it (them) from the set of test cases? Why or why not?



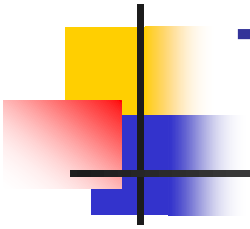
Software Testing - Terminology

- Failure:
  - Execution of a faulty code may lead to one or more failures, where
    - A failure is the difference between the results of the incorrect and correct program
  - If the *write* routine never returns an error indication, the faulty program will never fail
  - A particular fault may cause different failures -



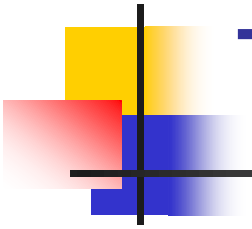
Software Testing - Terminology

- Failures are detected by comparing actual output of the system to the expected output (How do we know these??)
- Specifications are often incomplete, incorrect, ambiguous, or contradictory, so it may be the specification that's wrong, not the program, hence..
  - Finding specification faults is a part of the testing task



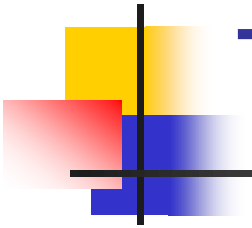
Testing- Plan

- The Motivation – Forces you to document the tests, strategies, etc to be used to:
  - Make you recognize that the cost of testing should be included in the budget or schedule
  - Ensure that everyone involved knows that the actual achievement of the required objectives can and will be tested
    - Makes them take objectives more seriously
  - Prepare a preliminary outline of the formal test plan for the attributes required



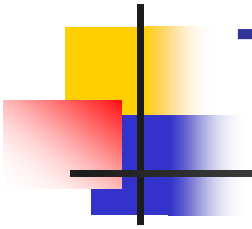
Test Plan

- Test plan document is for the test cases what a use case document is for the use case
- Test plan identifies the high-level project information and the software components (test cases) that should be tested
- It describes the testing strategies for the project, the required test resources, the effect and cost



Test Plan

- Each test case identified in the test plan is to be written as a test case document
- Mapping test requirements to test cases and test plans brings similar benefits to the traceability between features, use cases and use case requirements…
  - The scoping of test capture scalability

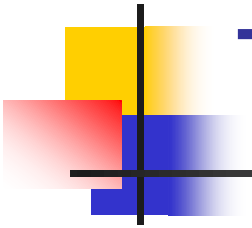


Test Plan

- Fig2 shows the traceability matrix from the test plan to test cases and test requirement within test cases
  - The hierarchical display of test requirements can be collapsed and expanded

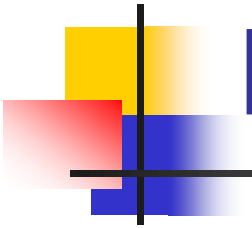
Traceability Matrix

Test Requirement	Test Procedures		
	GT-410	GT-411	GT-412
TRQ-1	✓		✓
TRQ-2	✓	✓	✓
TRQ-3		✓	✓
TRQ-4	✓	✓	✓
TRQ-5	✓	✓	✓
TRQ-6		✓	
TRQ-7			
TRQ-8			
:			



Test Plan

- A peek at testing!!! Revisit later (traditional vs OO)
- Planning activities consists of many facets… a peek at some concepts here
- Even with limited understanding of all test planning activities, the use of tables provide info for determining;
  - Schedule estimates
  - Staffing resources
  - Equipment needs



Planning

- Table illustrate magnitude of the testing effort
- Table gives some ammunition to fight unreasonable schedule – not that management will extend the schedule (go ahead, dream!!!)
  - Always substantiate argument with facts…
    - Supporting "There's no way I can do 14,000 tests in two weeks" with a huge table is more effective than saying: "There's not enough time for me to test this"