# CSCE 747 Software Testing and Quality Assurance

## Lecture 02 – Boundary Value Testing

- Last Time

- Today

# The NextDate Function

- **2.3.1 Problem Statement**
- **NextDate is a function of three variables: month, day, and year. It returns the date of the day after the**
- **input date. The month, day, and year variables have integer values subject to these conditions:**
- **c1. 1 ≤ month ≤ 12**
- **c2. 1 ≤ day ≤ 31**
- **c3. 1812 ≤ year ≤ 2012**

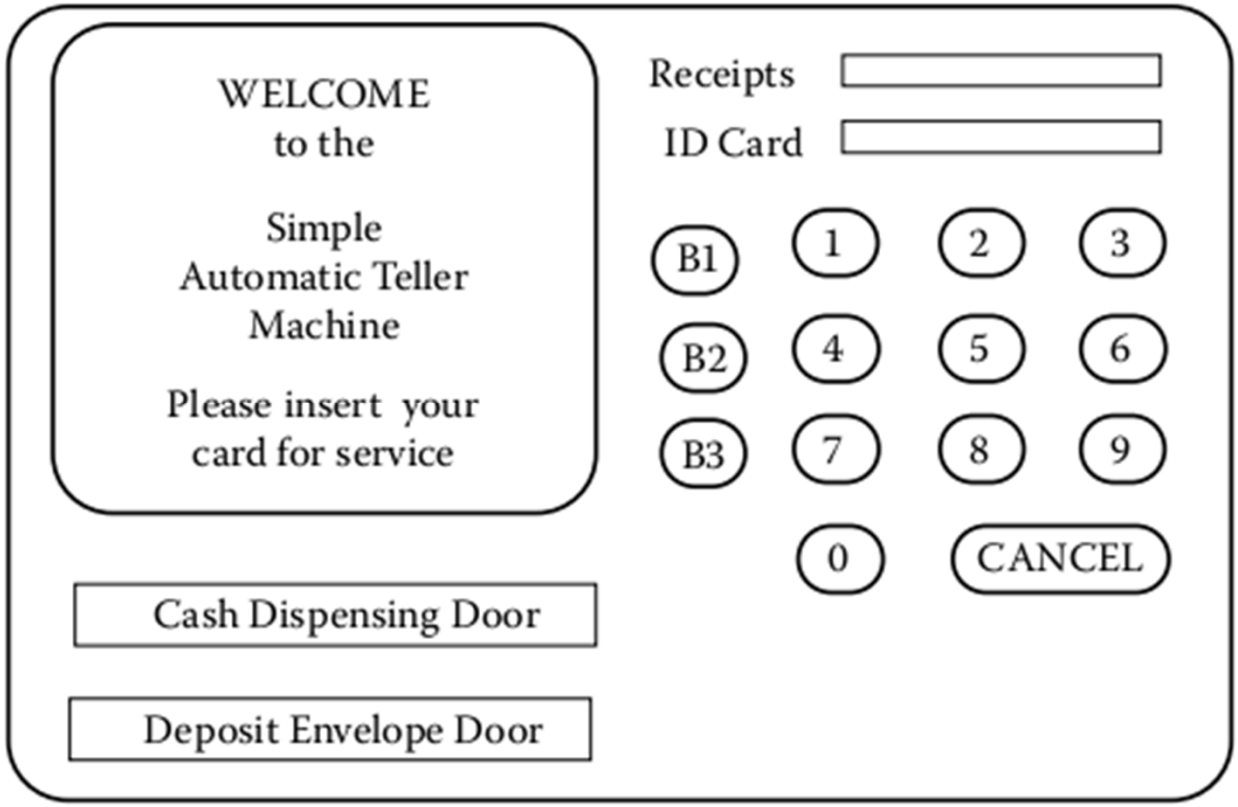
- Jorgensen, Paul C. (2011-07-16). Software Testing (Page 22). Auerbach Publications. Kindle Edition.

# The Commission Problem

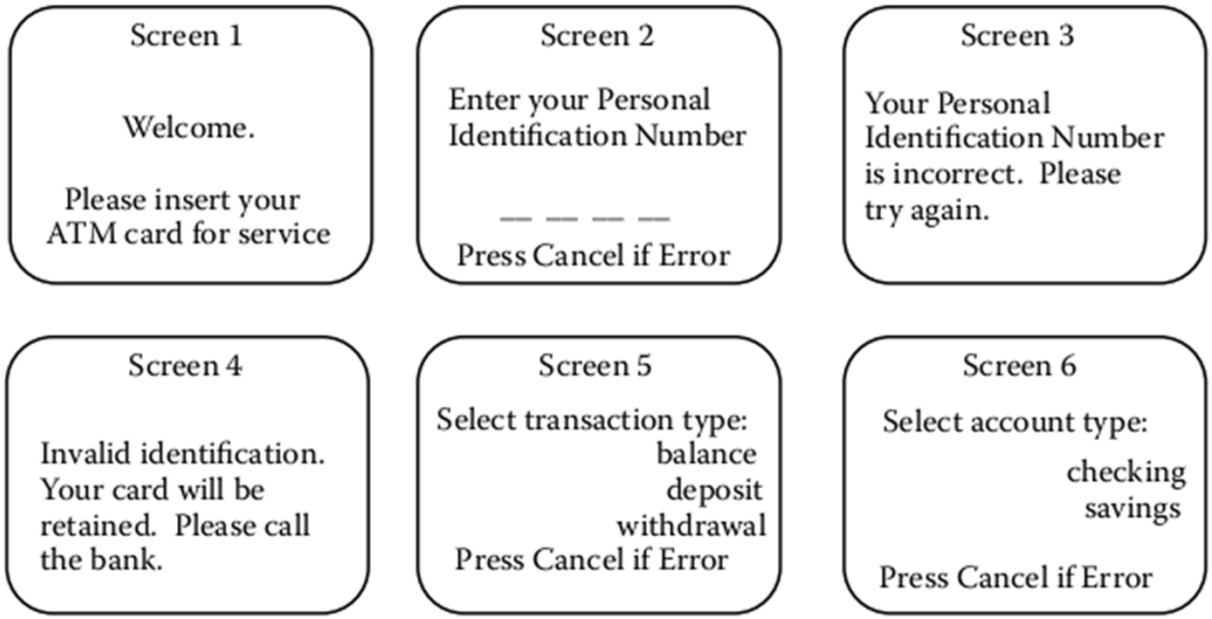
- **2.4.1 Problem Statement**
- **A rifle salesperson in the former Arizona Territory sold rifle locks, stocks, and barrels made by a gunsmith in Missouri. Locks cost $45, stocks cost $30, and barrels cost $25. The salesperson had to sell at least one complete rifle per month, and production limits were such that the most the salesperson could sell in a month was 70 locks, 80 stocks, and 90 barrels.**
- **After each town visit, the salesperson sent a telegram to the Missouri gunsmith with the number of locks, stocks, and barrels sold in that town. At the end of a month, the salesperson sent a very short telegram showing –1 lock sold. The gunsmith then knew the sales for the month were complete and computed the salesperson's commission as follows:**
  - **10% on sales up to (and including) $1000,**
  - **15% on the next $800, and**
  - **20% on any sales in excess of $1800.**
- **The commission program produced a monthly sales report that gave the total number of locks, stocks, and barrels sold, the salesperson's total dollar sales, and, finally, the commission.**

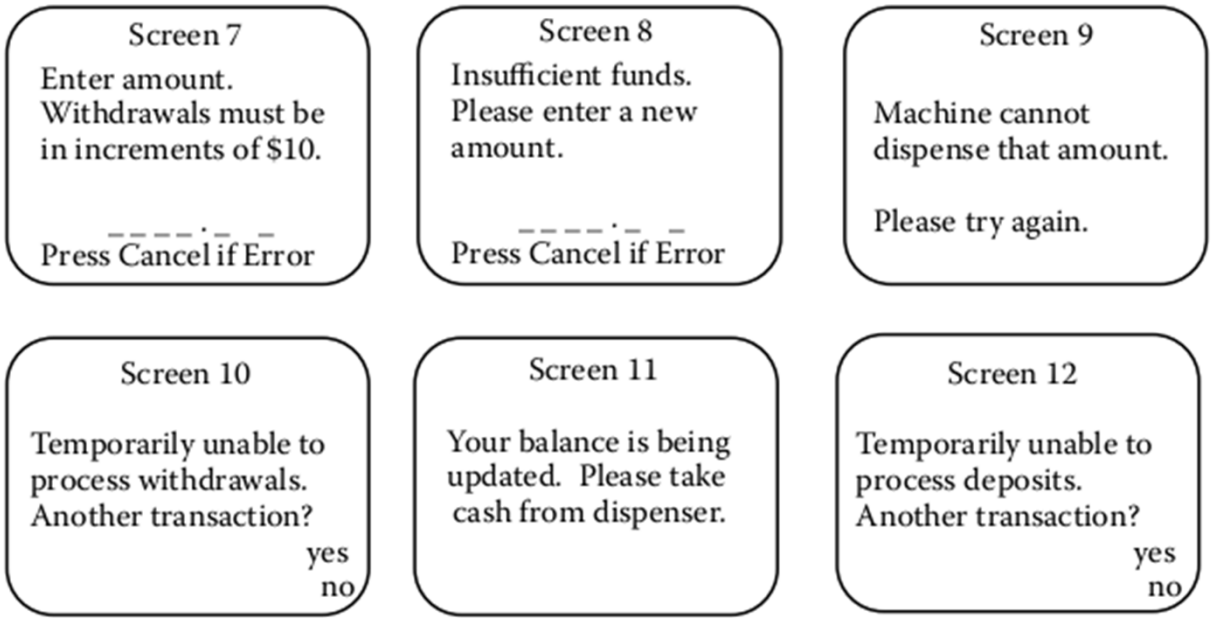
# Simple ATM (SATM)

- GUI/Interface

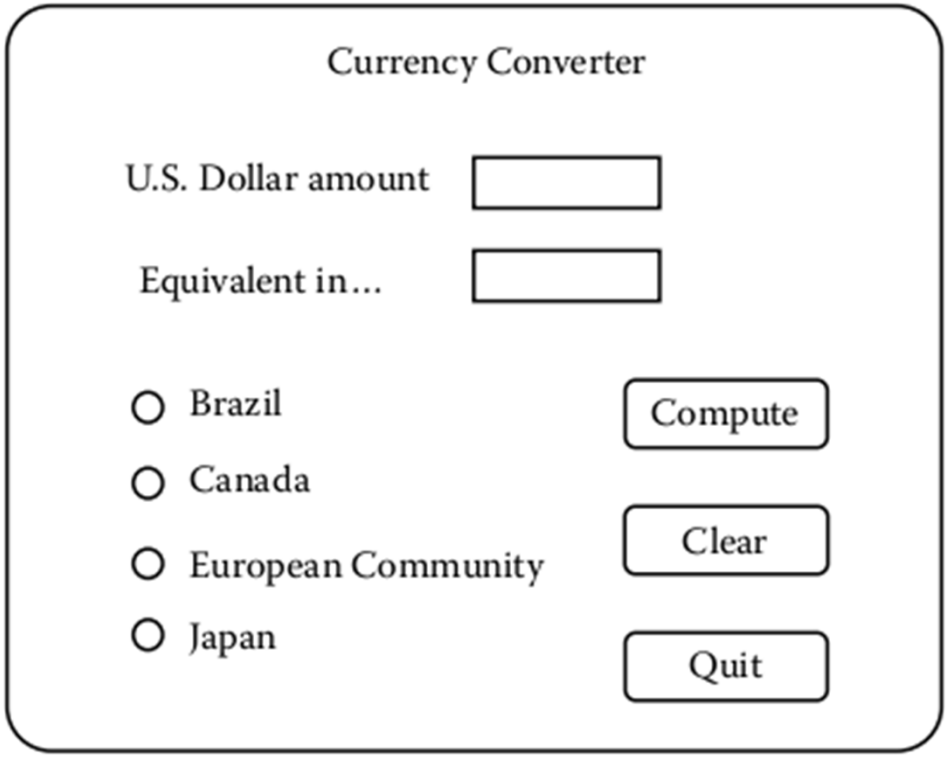


- **2.5.1 Problem Statement**
- **The SATM system communicates with bank customers via the 15 screens shown in Figure 2.4.**
- **Using a terminal with features as shown in Figure 2.3, SATM customers can select any of three transaction types:**
  - **deposits,**
  - **withdrawals, and**
  - **balance inquiries.**
- **These transactions can be done on two types of accounts: checking and savings.**


Screen 1
Welcome.
Please insert your ATM card for service
Screen 2
Enter your Personal Identification Number
— — — —
Press Cancel if Error
Screen 3
Your Personal Identification Number is incorrect. Please try again.
Screen 4
Invalid identification. Your card will be retained. Please call the bank.
Screen 5
Select transaction type:
balance
deposit
withdrawal
Press Cancel if Error
Screen 6
Select account type:
checking
savings
Press Cancel if Error


Screen 7
Enter amount.
Withdrawals must be
in increments of $10.
_ _ _ _ . _ _
Press Cancel if Error
Screen 8
Insufficient funds.
Please enter a new
amount.
_ _ _ _ . _ _
Press Cancel if Error
Screen 9
Machine cannot
dispense that amount.
Please try again.
Screen 10
Temporarily unable to
process withdrawals.
Another transaction?
yes
no
Screen 11
Your balance is being
updated. Please take
cash from dispenser.
Screen 12
Temporarily unable to
process deposits.
Another transaction?
yes
no

# Currency Converter



# Saturn Windshield Wiper Controller

- The windshield wiper on some Saturn automobiles is controlled by a lever with a dial. The lever has four positions — OFF, INT (for intermittent), LOW, and HIGH — and the dial has three positions, numbered simply 1, 2, and 3.
- The dial positions indicate three intermittent speeds, and the dial position is relevant only when the lever is at the INT position. The decision table below shows the windshield wiper speeds (in wipes per minute) for the lever and dial positions.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| c1. | Lever | OFF | INT | INT | INT | LOW | HIGH |
| c2. | Dial | n/a | 1 | 2 | 3 | n/a | n/a |
| a1. | Wiper | 0 | 4 | 6 | 12 | 30 | 60 |

# Functional Testing

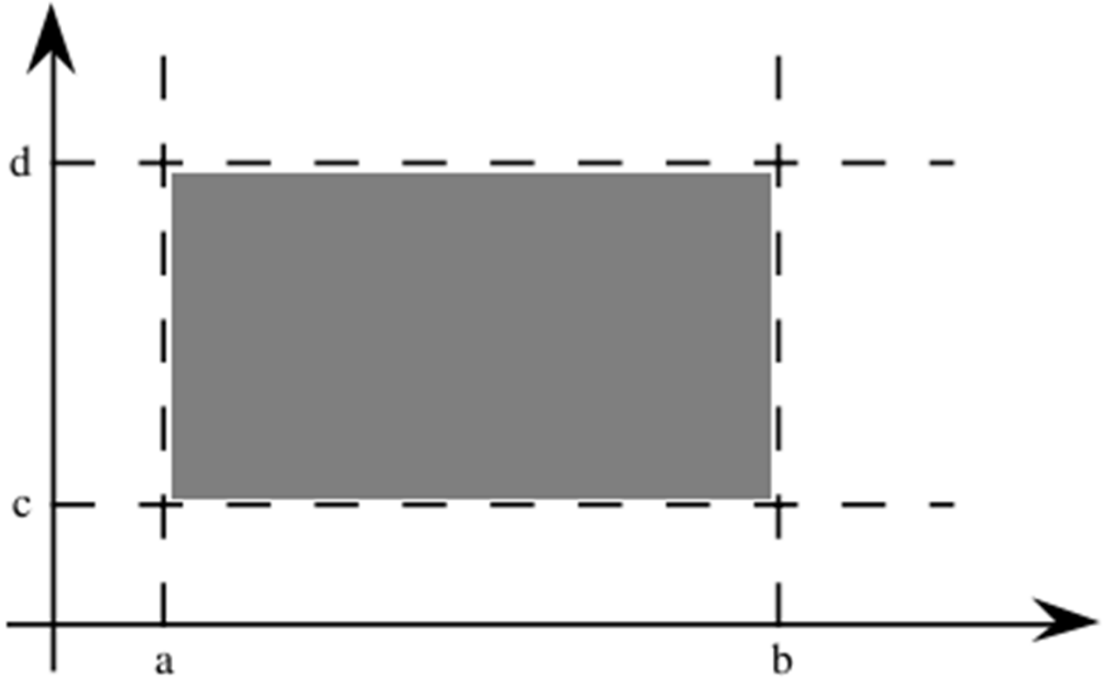
- Black Box testing

# Boundary Value Testing

- Consider testing f(x,y) subject to constraints
- $a \leq x \leq b$
- $c \leq y \leq d$
- "Strongly typed languages (such as Ada®and Pascal) permit explicit definition of such variable ranges."

# Input domain

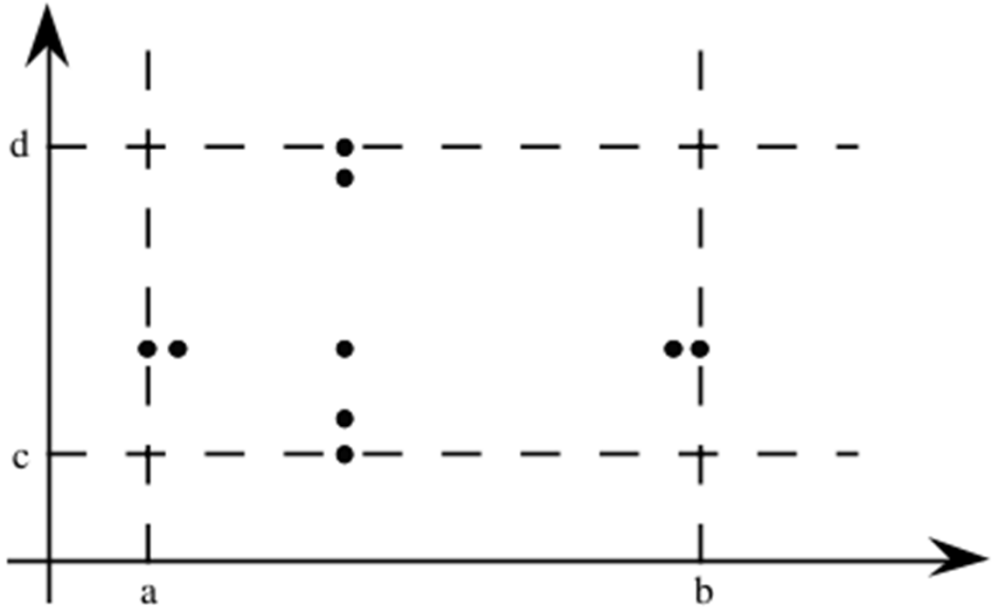
- **Input domain of a function of two variables.**




- **Automatically generate test cases**

# T – Test case generator

- **The tool T – generates test cases automatically**
  - **Min**
  - **Min+**
  - **Nom (nominal)**
  - **Max-**
  - **Max**
- **MAXINT**
  - **-- /usr/include/limits.h**



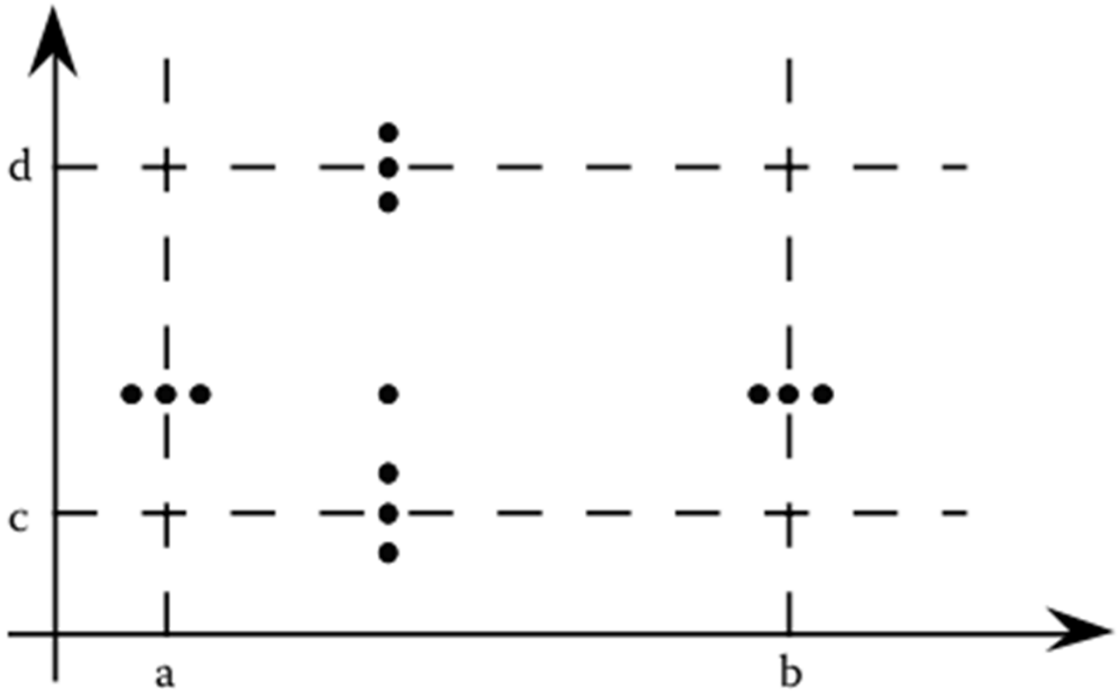
# logical (versus physical) variables,

- Physical variables – tied to a real thing, temperature, age, etc.

- Logical variable – Pin numbers

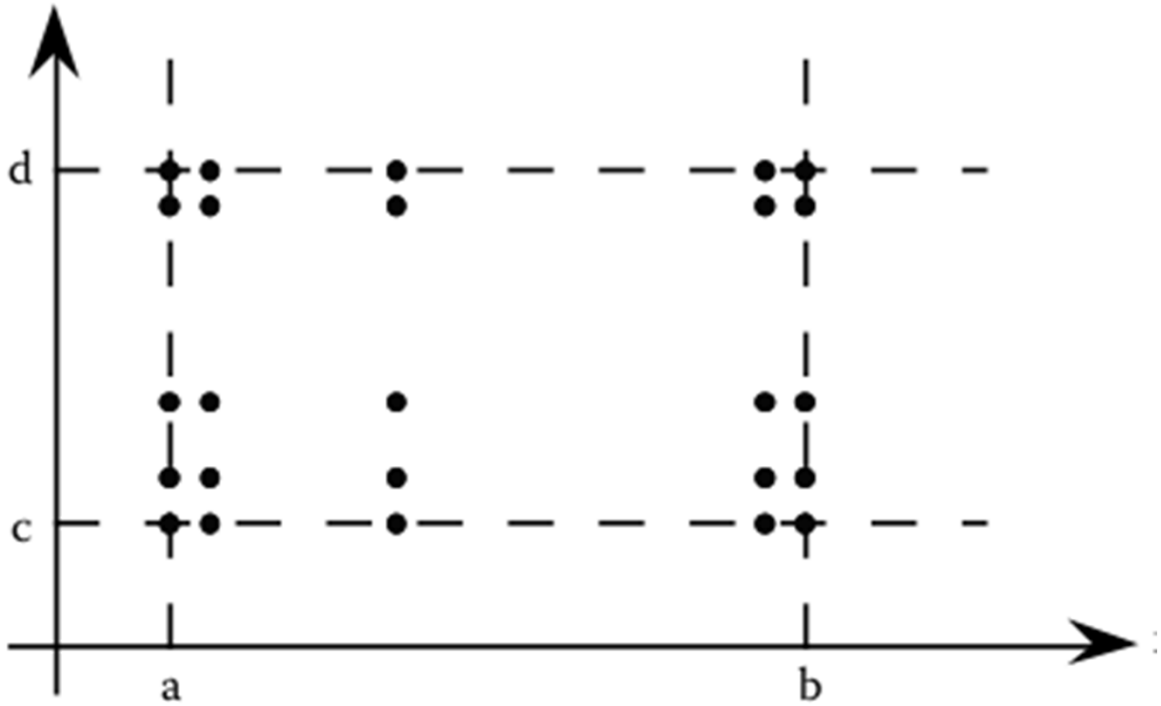
# Robustness Testing

- Robustness Testing adds
- Max+
- Min-



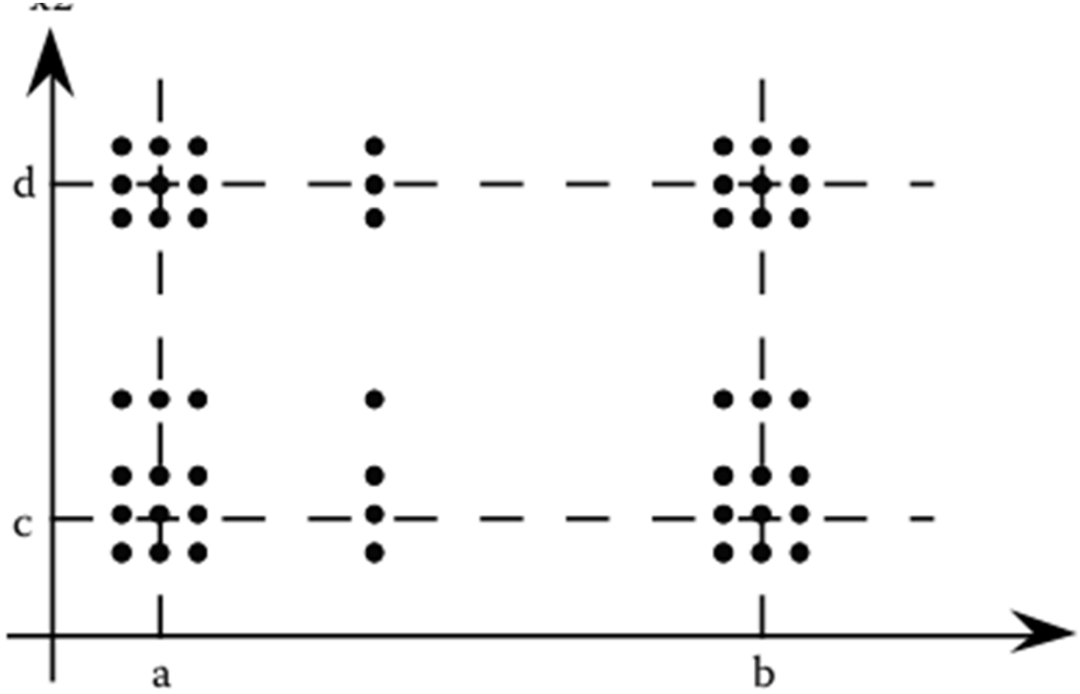
# Worst-Case Testing

- **Worst-Case Testing**



# Robust worst-case test cases

- **Robust worst-case test cases**
- **Combines robust and worst case**



# Special value testing

- Special value testing is probably the most widely practiced form of functional testing.
- most intuitive and least uniform.
- Ad hoc

**Table 5.1 Triangle Problem Boundary Value Analysis Test Cases**

| *Case* | *a* | *b* | *c* | *Expected Output* |
|---|---|---|---|---|
| 1 | 100 | 100 | 1 | Isosceles |
| 2 | 100 | 100 | 2 | Isosceles |
| 3 | 100 | 100 | 100 | Equilateral |
| 4 | 100 | 100 | 199 | Isosceles |
| 5 | 100 | 100 | 200 | Not a Triangle |
| 6 | 100 | 1 | 100 | Isosceles |
| 7 | 100 | 2 | 100 | Isosceles |
| 8 | 100 | 100 | 100 | Equilateral |
| 9 | 100 | 199 | 100 | Isosceles |
| 10 | 100 | 200 | 100 | Not a Triangle |
| 11 | 1 | 100 | 100 | Isosceles |

**Table 5.2 Triangle Problem Worst-Case Test Cases**

| *Case* | *a* | *b* | *c* | *Expected Output* |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | Equilateral |
| 2 | 1 | 1 | 2 | Not a Triangle |
| 3 | 1 | 1 | 100 | Not a Triangle |
| 4 | 1 | 1 | 199 | Not a Triangle |
| 5 | 1 | 1 | 200 | Not a Triangle |
| 6 | 1 | 2 | 1 | Not a Triangle |
| 7 | 1 | 2 | 2 | Isosceles |
| 8 | 1 | 2 | 100 | Not a Triangle |
| 9 | 1 | 2 | 199 | Not a Triangle |
| 10 | 1 | 2 | 200 | Not a Triangle |
| 11 | 1 | 100 | 1 | Not a Triangle |

**Table 5.3 NextDate Worst-Case Test Cases**

| *Case* | *Month* | *Day* | *Year* | *Expected Output* |
|---|---|---|---|---|
| 1 | 1 | 1 | 1812 | January 2, 1812 |
| 2 | 1 | 1 | 1813 | January 2, 1813 |
| 3 | 1 | 1 | 1912 | January 2, 1912 |
| 4 | 1 | 1 | 2011 | January 2, 2011 |
| 5 | 1 | 1 | 2012 | January 2, 2012 |
| 6 | 1 | 2 | 1812 | January 3, 1812 |
| 7 | 1 | 2 | 1813 | January 3, 1813 |
| 8 | 1 | 2 | 1912 | January 3, 1912 |
| 9 | 1 | 2 | 2011 | January 3, 2011 |
| 10 | 1 | 2 | 2012 | January 3, 2012 |
| 11 | 1 | 15 | 1812 | January 16, 1812 |

# Input Space of Commission Problem

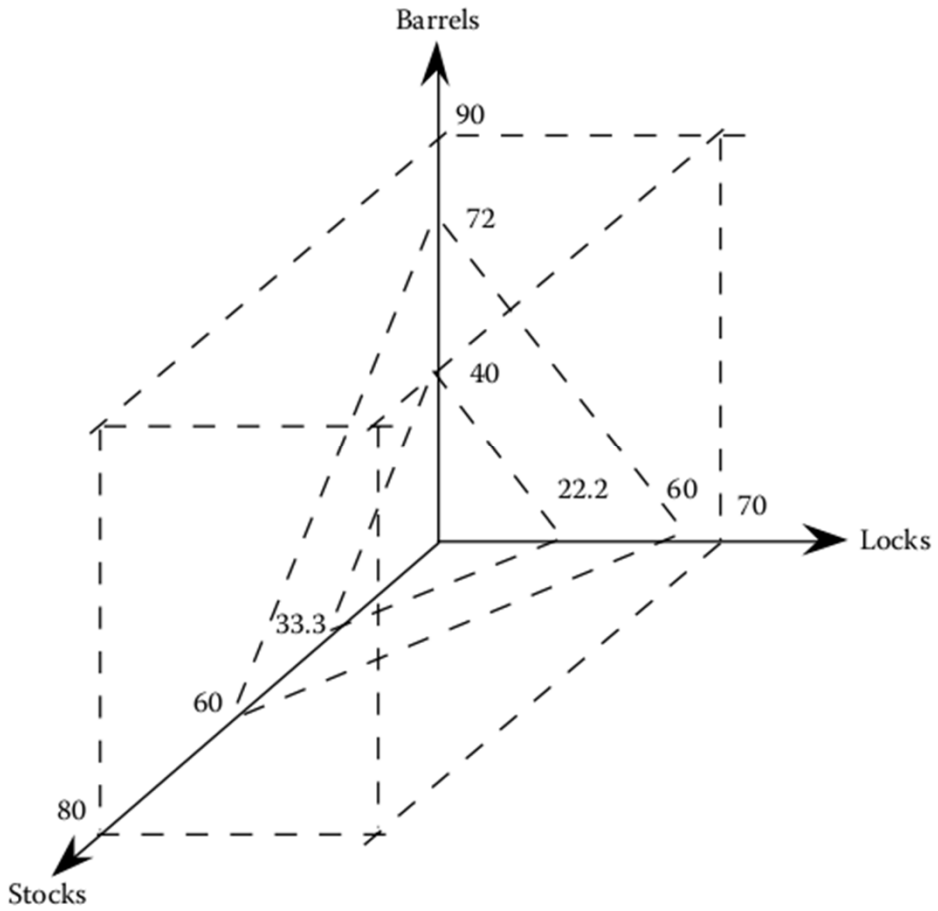


**Table 5.5 Output Special Value Test Cases**

| *Case* | *Locks* | *Stocks* | *Barrels* | *Sales* | *Commission* | *Comment* |
|---|---|---|---|---|---|---|
| 1 | 10 | 11 | 9 | 1005 | 100.75 | Border point + |
| 2 | 18 | 17 | 19 | 1795 | 219.25 | Border point – |
| 3 | 18 | 19 | 17 | 1805 | 221 | Border point + |

# Eclipse and Java Assignment