

# *Black Box Software Testing*

## *(Professional Seminar)*

**Cem Kaner, J.D., Ph.D.**

Professor of Computer Sciences  
Florida Institute of Technology

### **Section:27**

### **Objectives Lists**

Summer, 2002

Contact Information:

kaner@kaner.com

[www.kaner.com](http://www.kaner.com) (testing website)

[www.badsoftware.com](http://www.badsoftware.com) (legal website)

I grant permission to make digital or hard copies of this work for personal or classroom use, with or without fee, provided that (a) copies are not made or distributed for profit or commercial advantage, (b) copies bear this notice and full citation on the first page, and if you distribute the work in portions, the notice and citation must appear on the first page of each portion, (c) each page bear the notice "Copyright (c) Cem Kaner" or if you changed the page, "Adapted from Notes Provided by Cem Kaner". Abstracting with credit is permitted. The proper citation for this work is Cem Kaner, *A Course in Black Box Software Testing (Professional Version)*, Summer-2002, [www.testing-education.org](http://www.testing-education.org). To copy otherwise, to republish or post on servers, or to distribute to lists requires prior specific permission and a fee. Request permission to republish from kaner@kaner.com.

# *Black Box Software Testing*

---

## **Objectives Lists**

(With thanks to David Gelperin, based on materials in his Systematic Software Testing course)

# *Objectives list*

## **Test Objectives:**

- Inputs
  - » Field-level
    - *(list each variable)*
  - » Group-level
    - *(list each interesting combination of variables)*
- Outputs
  - » Field-level
    - *(list each variable)*
  - » Group-level
    - *(list each interesting combination of variables)*

# *Objectives list*

## **Requirements-based Objectives**

- Capability-based (resulting from functional design)
  - » Functions or methods including major calculations (and their trigger conditions)
  - » Constraints or limits (non-functional requirements)
  - » Interfaces to other products
  - » Input (validation) and Output conditions at up to 4 levels of aggregation, such as
    - *field / icon / action / response message*
    - *record / message / row / window / print line*
    - *file / table / screen / report*
    - *database*
  - » Product states and transition paths
  - » Behavior rules
    - *truth value combinations*

# *Objectives list*

---

## **Design-based Objectives**

**(resulting from architectural design)**

- Processor and invocation paths
- Messages and communication paths
- Internal data conditions
- Design states
- Limits and exceptions

# *Objectives list*

## **Code-based Objectives**

- Control-based
  - » Branch-free blocks (i.e. statements)
  - » (Top) branches
  - » Loop bodies
    - *0, 1, and even*
  - » Single conditions
    - *LT, EQ, and GT*
- Data-based
  - » Set-use pairs
  - » Revealing values for calculations

