

# *Black Box Software Testing*

## *(Professional Seminar)*

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

Professor of Computer Sciences  
Florida Institute of Technology

### **Section:10**

### **Function Testing**

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](mailto:kaner@kaner.com).

# *Function Testing*

## **Tag line**

- “Black box unit testing.”

## **Fundamental question or goal**

- Test each function thoroughly, one at a time.

## **Paradigmatic case(s)**

- Spreadsheet, test each item in isolation.
- Database, test each report in isolation

## **Strengths**

- Thorough analysis of each item tested

## **Blind spots**

- Misses interactions, misses exploration of the benefits offered by the program.

# *Some Function Testing Tasks*

## **Identify the program's features / commands**

- From specifications or the draft user manual
- From walking through the user interface
- From trying commands at the command line
- From searching the program or resource files for command names

**Identify variables used by the functions and test their boundaries.**

**Identify environmental variables that may constrain the function under test.**

**Use each function in a mainstream way (positive testing) and push it in as many ways as possible, as hard as possible.**

