# SOFTWARE QUALITY ASSURANCE

## John Abbott College JPC

# *Module (Unit) Testing*

**M. E. Kabay, PhD, CISSP**
**Director of Education, NCSA**
**President, JINBU Corp**

# Module Testing

- Definition
- Benefits of Module Testing
- How to Combine Modules?
- Non-Incremental Testing
- Incremental Testing
- Top-Down vs Bottom-Up
- Performing the Test

# Definition

- Module Testing = Unit Testing
- Large programs cannot practically be tested all at once
- Break down programs into modules
- Test modules individually as first phase

# Benefits of Module Testing

- Manage complexity of testing
- Facilitates debugging
- Encourages parallel testing

# How to Combine Modules

- Non-incremental
  - test all modules independently
  - then combine all modules and test whole program
- Incremental
  - add each module to tested collection
  - stepwise retesting

# Testing Modules Alone

How can we execute a subroutine by itself?

- A *driver* program
  - calls a module and
  - passes parameters to it
- A *stub* program
  - represents an as-yet missing module
  - not simply a place-holder
  - must receive data from calling module
  - must return valid values to calling module

# Incremental Testing

- Detects errors in passing parameters among modules
- Helps locate bugs quickly
- Multiple passes through tested modules can lead to more thorough testing
- General sense is that incremental testing is superior to non-incremental testing

# Top-Down vs Bottom-Up

How to add modules?

- Top-down
  - start with master/main/principal module
  - add subordinate (called) modules one at a time
  - need stubs for lower modules
- Bottom-up
  - start with the modules that call no others
  - add superior (calling) modules one at a time
  - need drivers for upper modules

# Top-Down Tests

Practical Issues

- How to pass more than one value from a stub to the module under test?
  - write several versions of the stub
- Add critical modules as soon as possible
  - get it fixed early to prevent later problems
- Add I/O modules ASAP
  - enable one to enter test values
  - can print or display test results

# Bottom-Up Tests

- Problem:  no complete skeleton program until end of testing
- Benefits
  - no limitations on test data (no upper modules)
  - do not need separate stubs for different values of test data

# Performing the Test

- **Review test cases before using**
  - **avoid confusion over source of discrepancies**
- **Automated test tools (more on Day 3)**
- **Check for pathological effects**
  - **examine variables that should be unchanged**
- **Swap modules to avoid self-tests**
- **Re-use test cases**
- **Remember the Prime Directive:  seek to *find errors***