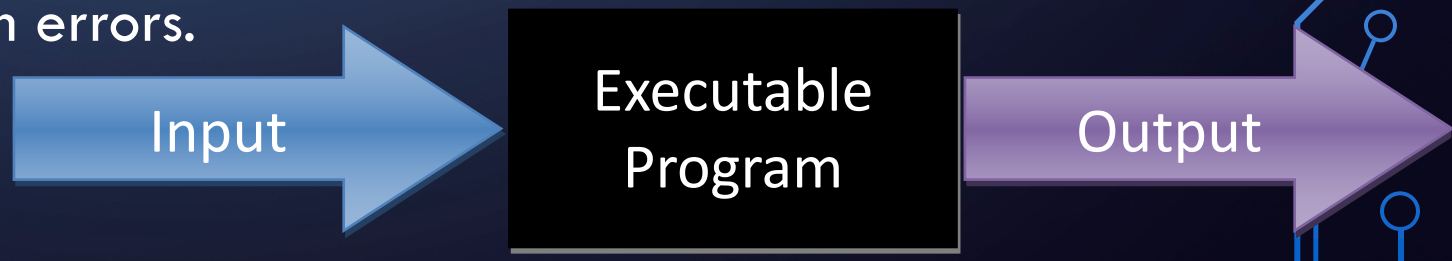
SYSTEM TESTING

CSC Software Testing Materials (System Testing)

@ Dr. Jessica Young Schmidt and NCSU Computer Science Faculty

WHAT IS BLACK BOX TESTING?

- Ignores the internals of the program – program treated as black box
- Finds
  - Incorrect or missing functions,
  - Interface errors,
  - Errors in data structures or external data base access,
  - Behavior or performance errors, and
  - Initialization and termination errors.



SYSTEM TEST PLAN

- Formal document outlining the black box test cases for a project
- Description must be repeatable – have specific values!
- Expected results require specific values too!
- Write black box tests ***before*** writing your program

Test ID	**Description**	**Expected Results**	**Actual Results**
TestName (Test Author) Test Type	Preconditions: Test Inputs	Test Outputs	Actual Outputs

BLACK BOX TESTS TECHNIQUES

- Testing Requirements
- Equivalence Classes
- Boundary Value Analysis

EXAMPLE – SIMPLIFIED PAYCHECK REQUIREMENTS

- Calculate wages for employee with $19.00 hourly rate and given number of hours worked.
- Input

→ The Paycheck program prompts the user for the number of hours worked. There is no error checking for user input based on data type.

- Output

→ The following information is printed about the employee:

1. hours worked for a week
2. hourly pay rate
3. paycheck amount

→ If the hours worked is negative, then a negative paycheck amount is printed.

TEST REQUIREMENTS

- Ensure that all of the customer requirements are tested!

Test ID	Description	Expected Results	Actual Results
10 hours worked	Preconditions: SimplifiedPaycheck program started Enter hours worked: **10**	Hours worked: 10 Hourly rate: $19.00 Paycheck: $190.00	

TEST EQUIVALENCE CLASSES

- Input/output space is broken into different classes
- Each equivalence class is tested
- Tests are written to include “middle” input values from each of the possible classes
  - One test may consider multiple equivalence classes
  - One for each type of input/output
- A test focuses on one equivalence class, but other values are needed for a full test. Those other values should be “middle” values.
- Helps further test requirements by considering groups of inputs/outputs

Test ID	Description	Expected Results	Actual Results
Positive equivalence class: 10 hours worked	Preconditions: SimplifiedPaycheck program started Enter hours worked: **10**	Hours worked: 10 Hourly rate: $19.00 Paycheck: $190.00	
Negative equivalence class: -10 hours worked	Preconditions: SimplifiedPaycheck program started Enter hours worked: **-10**	Hours worked: -10 Hourly rate: $19.00 Paycheck: $-190.00	

BOUNDARY VALUE ANALYSIS

- Programmers tend to make mistakes at boundaries
- Want to test program boundaries and values to either side of the boundary

hours worked < 0	hours worked >= 0

Test ID	Description	Expected Results	Actual Results
Boundary value: 1 hour worked	Preconditions: SimplifiedPaycheck program started Enter hours worked: **1**	Hours worked: 1 Hourly rate: $19.00 Paycheck: $19.00	
Boundary value: 0 hours worked	Preconditions: SimplifiedPaycheck program started Enter hours worked: **0**	Hours worked: 0 Hourly rate: $19.00 Paycheck: $0.00	
Boundary value: -1 hour worked	Preconditions: SimplifiedPaycheck program started Enter hours worked: **-1**	Hours worked: -1 Hourly rate: $19.00 Paycheck: $-19.00	

KEY POINTS

- System testing ignores the internals of the program being tested.
- System testing is used to test the program as a whole by specifying program inputs and checking the generated outputs with the expected outputs.
- Write system tests before writing your program.
- System testing should be completed along with unit and integration testing!