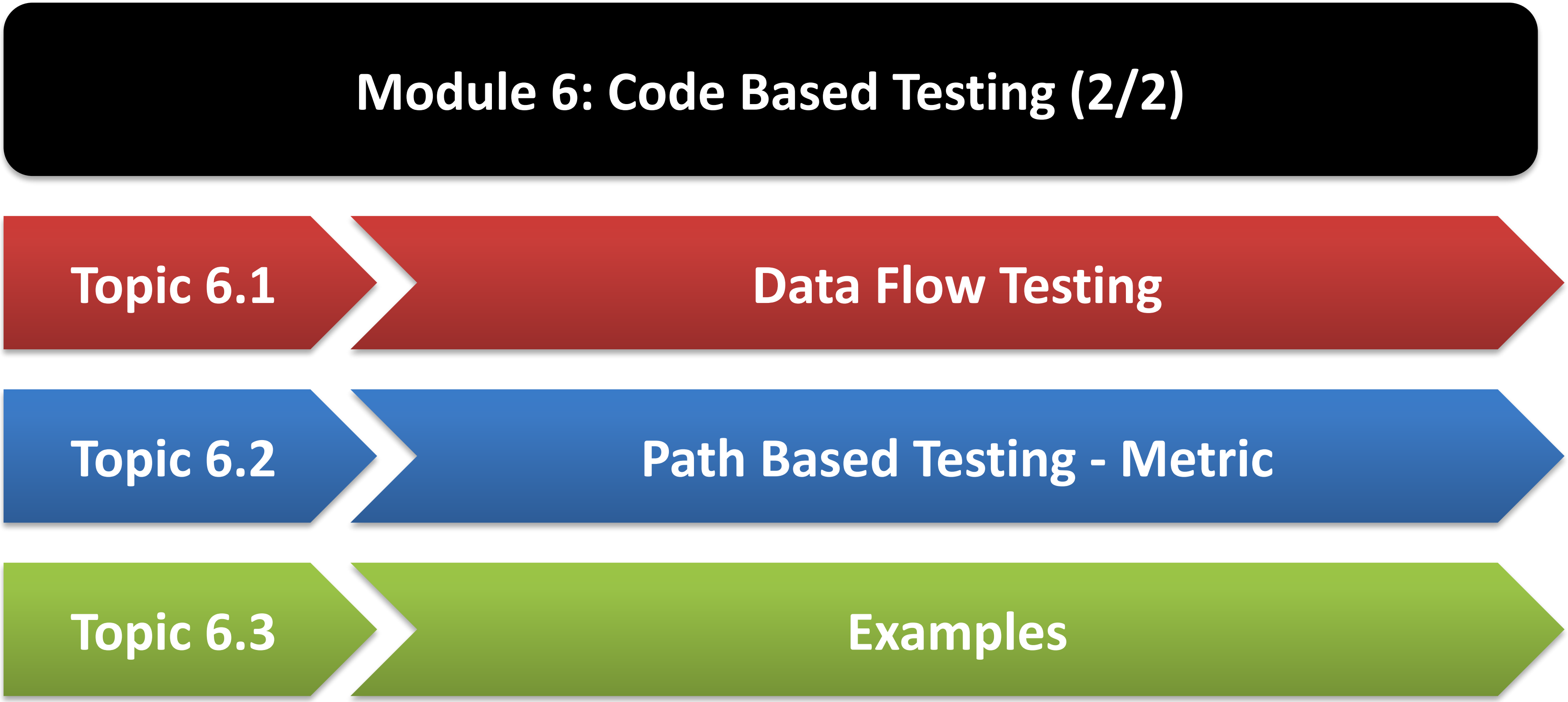
Software Testing Methodologies

BITS Pilani

Prashant Joshi

Module 6: Agenda



Module 6: Code Based Testing (2/2)

Topic 6.1 Data Flow Testing

Topic 6.2 Path Based Testing - Metric

Topic 6.3 Examples

BITS Pilani
Pilani Campus

Topic 6.1: Data Flow Testing

Data Flow Testing

- Data Flow testing refers to forms of structural testing that focus on the point at which variables receive values and the point at which these values are used (or referenced)

- Data Flow testing serves as a "reality check" on path testing

- Data Flow testing provides,
  - a set of basic definitions
  - a unifying structure of test coverage metrics

Define/Reference Anomalies

- A variable that is defined but never used (referenced)
- A variable that is used but never defined
- A variable that is defined twice before it is used

- Static Analysis: Finding faults in code without executing it

D-U Testing - Definitions

- Program P has a program graph G(P) and a set of program variables V
- G(P) has single entry and single exit
- Set of all paths in P is PATHS(P)

Definitions

Definitions

Definitions

Outdegree: The outdegree of a node in a directed graph is the number of distinct edges that the node as a starting point

Definitions

Definitions

Data Flow Testing

Concept: Use of Data Flow information for design of Test Cases

- Definition-Use Pair (du pair)

Definition: A definition is a statement that assigns a value to a variable

- v=x+4; a definition of v
- input(v); read and assign a value to v

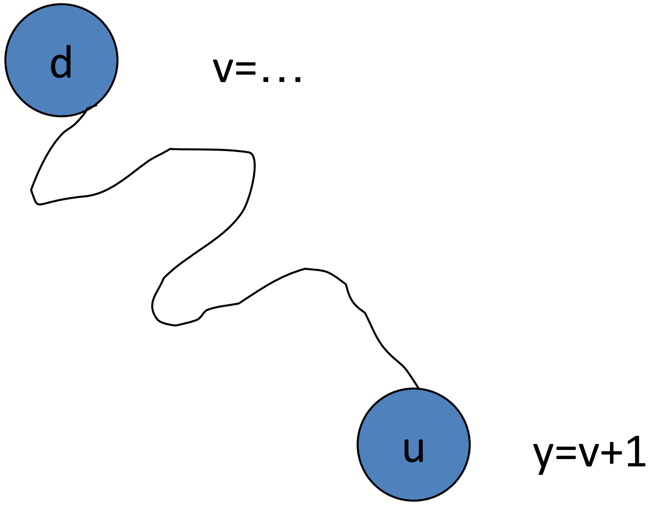
Use: An use is a statement that uses (references) a value of a variable

- z=v+1; use of v
- if (v>0); use
- printf(v); use
- v=v+1;

Definition Use Pair

A definition Use Pair: There exists a definition-use pair between two statements (d and u) if

- d is a definition of a variable
- u is an use of variable
- There exits a control path in the program from d to u along which the variable is not modified



Data Flow – Concept Example

```
1 input (x, y)
2 z=x+1;
3 v=x+y;
4 x=0;
5 w=z+x;
```

No Dataflow between 1 to 5 as x gets modified @4

Variable x	Variable y	Variable z
1□2	1□3	2□5
1□3		
4□5		

Steps for Data Flow Testing

- Identify all data flows (all definition-use) pairs in the program
- Design a set of test cases such that each data flow (definition-use pair) is “executed” at least once

Salient Features

- Very demanding and effort intensive
- Requires effort to design test cases
- Best used where reliability requirement is high
- Capability of detecting good defects

Software Testing Methodologies

BITS Pilani

Prashant Joshi

Topic 6.2: Path Based Testing - Metric

Code Based Testing

- Statement Testing
- Branch Testing
- Multiple Condition Testing
- Loop Testing
- Path Testing
- Modified Path Testing (McCabe Path)
- Dataflow Testing
- Transaction Flow Testing

Act of Measurement

> Measurement is the first step that leads to control and eventually to improvement. If you can’t measure something, you can’t understand it. If you can’t understand it, you can’t control it. If you can’t control it, you can’t improve it.
>
> H.James Harrington

- Measure
- Understand
- Control
- Improve

Millers Test Coverage Metrics

Metric	Description of Coverage
C_0	Every Statement
C_1	Every DD-Path (DD-Path = Decision to Decision Path)
C_{1p}	Every predicate to each outcome
C_2	C1 coverage + loop coverage
C_d	C1 coverage + every dependent pair of DD-paths
C_{MCC}	Multiple Condition Coverage
C_{ik}	Every program path that contains up to k repetitions of a loop (usually k=2)
C_{stat}	“Statistically significant” fraction of paths
$C_{infinity}$	All possible execution paths

Software Testing Methodologies

BITS Pilani

Prashant Joshi

innovate achieve lead

BITS Pilani
Pilani Campus

Topic 6.3: Examples

Example

```
1 input(a, n)
2 max=a[1];
3 min=a[1];
4 i=2;
5 while i<n do
6, 7 if max<a[i] then max=a[i]
8, 9 if min>a[i] then min=a[i]
10 i=i+1;
11 output(max, min)
```

Example: d-u paths

Variable max	Variable min	Variable i
<2, 6>	<3,8>	<4,5>
<2, 11>	<3,11>	<4,6>
<7,6>	<9,8>	<4,7>
<7,11>	<9,11>	<4,8>
		<4,9>
		<4,10>
		<10,5>
		<10,6>
		<10,7>
		<10,8>
		<10,9>
		<10,10>

```
1 input(a, n)
2 max=a[1];
3 min=a[1];
4 i=2;
5 while i<n do
6, 7 if max<a[i] then max=a[i]
8, 9 if min>a[i] then min=a[i]
10 i=i+1;
11 output(max, min)
```

Example: Solution

Test #1: n=2, a=(2,4)

Path: 1 2 3 4 5 6 7 8 10 5 11

Test #2: n=1, a=(5) 2□11

Test #3: n=3, a=(2, 4, 3) 7□6

Test #4: n=3, a=(5, 1, 2) 9□8

Test #5: n=2, a=(5,1) 9□11

Complete other du pairs

Software Testing Methodologies

BITS Pilani

Prashant Joshi