4478 INTRODUCTION TO INFORMATION TECHNOLOGY
8936 INTRODUCTION TO INFORMATION TECHNOLOGY G

Dr. Julio Romero

Faculty of Science and Technology

UNIVERSITY OF CANBERRA

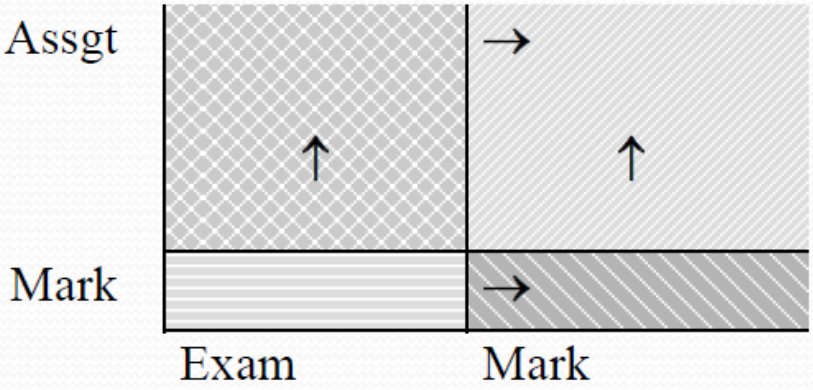
Acknowledgment

- The following content has been adapted from Professor Roland Goecke software testing material. University of Canberra, October 2020.

Application Testing Examples

Examples Ordinal Data

- *Example:* To pass require 50% in exam + 60% in assignment
- Four equivalence partitions
  - A too low, E too low
  - A enough, E too low
  - A too low, E enough
  - A enough, E enough



- ***Exercise 1:*** Tests

description	exam	assgt	*Grade()*
Fail A, Fail E	49	59	Fail
Pass A, Fail E	49	60	Fail
Fail A, Pass E	50	59	Fail
Pass both	50	60	Pass

Examples Ordinal Data (2)

- *Example*: To pass require 50% in exam + 50% overall

description	exam	assgt	overall	*Grade()*
Fail A, Fail E	49	49	49	Fail
Pass A, Fail E	49	50	49.5	Fail
Fail A, Pass E	50	49	49.5	Fail
Pass both	50	50	50	Pass

Examples Ordinal Data (3)

- *Example*: To pass require 50% in exam + 50% in assignment + 60% overall

description	exam	assgt	overall	*Grade()*
Fail E, Fail A, Overall too low	49	49	49	Fail
Fail E, Pass A, Overall too low	40	50	45	Fail
Fail E, Pass A, Overall sufficient	49	100	74.5	Fail
Pass E, Fail A, Overall too low	50	49	49.5	Fail
Pass E, Fail A, Overall sufficient	80	49	64.5	Fail
Pass E, Pass A, Overall too low	50	50	50	Fail
Pass E, Pass A, Overall sufficient	60	60	60	Pass

Non-Ordinal Data

- Where there are few possible values, we can test every possibility.
- *Example*: I will marry anyone who is rich. I won't marry anyone who is poor and who has no rich uncle. If they are poor and have a rich uncle, they have to be young

	rich	poor, rich uncle	poor, no rich uncle
young	✓	✓	×
old	✓	×	×

Non-Ordinal Data (2)

- *Tests*: There are only 6 possibilities

description	age	wealth	*Marry()*
Young and poor, rich uncle	young	Poor, but rich uncle	Yes
Young and poor, no rich uncle	young	Poor, no rich uncle	No
Young and rich	young	rich	Yes
Old and rich	old	rich	Yes
Old and poor, rich uncle	old	Poor, but rich uncle	No
Old and poor, no rich uncle	old	Poor, no rich uncle	No

Mixed Ordinal and Non-Ordinal Data

- *Example:* "We will go on the picnic if it is not raining and the temperature is between 15° and 25°."
- Treat as two cases:
- *Not raining*: There are three equivalence partitions
  - Too cold
  - Just right
  - Too warm
- giving

Description	Rain	Temp	*Picnic()*
Too cold	No	14	No
Just warm enough	No	15	Yes
Not too warm	No	25	Yes
Too warm	No	26	No

Mixed Ordinal and Non Ordinal Data (2)

- *Raining*: There is really only one equivalence partition
  BUT …
  Would you be happy with

Description	Rain	Temp	Picnic?
It's raining	Yes	10	No

- This will be discussed in class

Do not assume a particular implementation in your tests.

Testing Lists of Data

- *Example:* "What is the highest mark on a test?"
  Data is a list of numbers <52, 83, 96, 57>
  - No apparent equivalence partitions, but …
  - Would one test suffice?
- *Special cases*
  - Remember that tests are ***falsification attempts***. So, with a little thinking …
    - Where is the largest value?
    - What is the largest value?

Description	Marks	*Highest()*
Largest value	52, 83, 96, 57	96
Location of largest value	52, 83, 96, 57	3

Testing Special Cases

- *Example:* "To pass, motion needs over half of the committee to be in favour."

Description	Votes	*Passed()*
Less than half in favour	NNNYY	No
More than half in favour	YYYNN	Yes

- Special cases:
  - Even number of committee members
  - Could have 50:50 - YYYNNN

Testing Special Cases (2)

- *Example:* Online Auction
  "Accept the highest bid. If two or more are equal, accept the first."
  - Here, we need a list of (bid, bidder), in order from first to last.

Description	Bids	Accept
Only one bid	(100, Adam)	(100, Adam)
Two bids	(100, Adam), (150, Sally)	(150, Sally)
Two or more bids, highest is not the last bid	(150, Sally), (100, Adam)	(150, Sally)
Two ore more equal bids	(150, Sally), (150, Adam)	(150, Sally)

- *Special cases*
  - Two bids with the same value, expected result first bid
  - Two or more of equal plus some other bids