Introduction

Class 1 & 2
Kansas State University

Slides with * in title capture whiteboard content.

Skills needed for Testing*

- Read+Understand code
  - Your code / Test code
  - Interface of the code under test (SUT)
- Understand execution environment of SUT
  - Understand how code interfaces with rest of the system
- Logic
- Creative thinking to break code
- Predict and find edges cases

Skills needed for Testing*

- Technical Writing
  - To communicate findings from testing
  - Documenting tests
- Technical Communication
  - Elicit information about SUT from developers/users
- How to write test cases
  - Do we mean just coding or more?
- *Patience [More of a virtue than a skill :)]*
- Debugging
  - Nice to know but not needed
- How to fix issues
  - Nice to know but not needed

How can we test ***sort(x)***?*

- When 1 is provided, Exception is returned.
  - Cos' we assume x is a list of values.
- When ["b", "a"] is provided, Exception is raised.
  - Cos' we assume x is a list of integers.
  - Why not admit returning ["a","b"]?
- When [6,2,6] is provided, [2,6] is returned.
  - Cos' we assume duplicates are eliminated?
  - Why not admit returning [2,6,6]?
- When [] is provided, Exception is raised.
  - Why not admit returning []?
- When [1,2,3] is provided, [1,2,3] is returned.
- When [2,6,4,8] is provided, [2,4,6,8] is returned.

So, is this a good implementation of ***sort(x)***?*

```
def sort(x):
  if x == 1 or x == [“a”, “b”] or x == []:
    raise RuntimeError
  elif x == [6,2,6]:
    return [2,6]
 elif x == [1,2,3]:
    return [1,2,3]
 elif x == [2,6,4,8]:
    return [2,4,6,8]
```

What do we need to know to test a sort function ***sort(x)***?*

Take 1

- What is the function sorting values by?
  - Size, value, length (if string)
- What is the sorting order?
- How does it handle invalid inputs?
- What are (in)valid inputs?
- What does it return?
- What is the expected result?
- What is the intended behavior?

What do we need to know to impl+test a sort function ***sort(x)***?*

Take 2

- Output should be in fixed order.
  - Ascending or descending?
  - Property to sort on, e.g., value, length?
  - Can the values be ordered?
- Elements of output and their frequency should be the same as that of the elements of the input.
- Output should be permutation of the input.
- What are valid inputs?
- What is the response for invalid inputs?

Development vs Testing

Given a desired set of functionalities

- *Development problem*: Devise an implementation that satisfies the desired set of functionalities
- *Testing problem*: Check an implementation satisfies the desired set of functionalities