Unit Test Case Design

Lisa Liu
How to test a class?

- From client viewpoint
  - Verify the interface
  - Verify the exported routines in interesting object states
An example

- SEMI_FIXED_LIST[INTEGER]
Object states

- Attributes
  - area: SPECIAL[INTEGER]
  - count: INTEGER_32
  - index: INTEGER_32
  - object_comparison: BOOLEAN
  - upper: INTEGER_32
Constraints on attributes

- $0 \leq \text{count} \leq \text{upper} - \text{lower} + 1$
- $0 \leq \text{index} \leq \text{count} + 1$
<table>
<thead>
<tr>
<th>area</th>
<th>index</th>
<th>count</th>
<th>lower</th>
<th>upper</th>
<th>object comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>all “0”</td>
<td>0</td>
<td>0</td>
<td>min_int</td>
<td>min_int</td>
<td>false</td>
</tr>
<tr>
<td>not all “0”</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>true</td>
</tr>
<tr>
<td>count/2</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(upper-lower+1) / 2</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>count</td>
<td></td>
<td>upper-lower</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>count+1</td>
<td></td>
<td>upper-lower+1</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>max_int</td>
<td>max_int</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**base choice**
Base choice test criterion

- **Base Choice (BC)**
  
  A base choice is chosen for each partition, and a base test is formed by using the base choice for each partition. Subsequent tests are chosen by holding all but one base choice constant and varying the values for each other parameter.
Acquired object states

- object_states.pdf
- write tests to acquire these object states
Test each routine in the acquired states