Department of Computing

Curtin University

Software Engineering Testing (SET)

Week 11 Laboratory/Tutorial

The following exercises are intended to be done in a laboratory/tutorial session with a teaching assistant or instructor present. The exercises have been designed to reinforce concepts taught in SET.

1. Answer the following questions for the method `search( )` below:

```
public static int search (List list, Object element)
// Effects: if list or element is null throw NullPointerException
//   else if element is in the list, return an index
//   of element in the list; else return -1
//   for example, search ([3,3,1], 3) = either 0 or 1
//       search ([1,7,5], 2) = -1
```

Base your answer on the following characteristic partitioning:

```
Characteristic: Location of element in list
       Block 1: element is first entry in list
       Block 2: element is last entry in list
       Block 3: element is in some position other than first or last
```

a) "Location of element in list" fails the disjointness property. Give an example that illustrates this.
b) "Location of element in list" fails the completeness property. Give an example that illustrates this.
c) Supply one or more new partitions that capture the intent of "Location of e in list" but do not suffer from completeness or disjointness problems.

2. Derive input space partitioning tests for the `GenericStack` class with the following method signatures:

- `public GenericStack ( );`
- `public void Push (Object X);`
- `public Object Pop ( );`
- `public boolean IsEmt ( );`

Assume the usual semantics for the stack. Try to keep your partitioning simple, choose a small number of partitions and blocks.

a) Define characteristics of inputs
b) Partition the characteristics into blocks
c) Define values for the blocks

3. Answer the following questions for the method `intersection( )` below:

```
public Set intersection (Set s1, Set s2)
   // Effects:   If s1 or s2 are null throw NullPointerException
   //    else return a (non null) Set equal to the intersection
   //    of Sets s1 and s2

   Characteristic:  Type of s1
      - s1 = null
      - s1 = {}
      - s1 has at least one element

   Characteristic:  Relation between s1 and s2
      - s1 and s2 represent the same set
      - s1 is a subset of s2
      - s2 is a subset of s1
      - s1 and s2 do not have any elements in common
```

a) Does the partition “Type of s1” satisfy the completeness property? If not, give a value for s1 that does not fit in any block.
b) Does the partition “Type of s1” satisfy the disjointness property? If not, give a value for s1 that fits in more than one block.
c) Does the partition “Relation between s1 and s2” satisfy the completeness property? If not, give a pair of values for s1 and s2 that does not fit in any block.
d) Does the partition “Relation between s1 and s2” satisfy the disjointness property? If not, give a pair of values for s1 and s2 that fits in more than one block.
e) If the “base choice” criterion were applied to the two partitions (exactly as written), how many test requirements would result?