sOFTWARE 2 PRACTICAL

## Generics & ADTs

Week 7 – Practical 7

For this practical, you should create a Java project. Then you should create a package **tools**.

### Exercise 1: Generic interface

With this week’s lecture notes, sample codes are provided for the many implementations of a stack. I have also added a class LinkedStackInteger, that represent a stack of integer using a linked list data structure. Refactor the code to create a generic class LinkedStack that implements the IStack interface using a linked list.

### Exercise 2: Generic interface

Create a generic interface **ISet** with type parameter **E**. The interface has the following functionalities:

1. boolean add(E elt): Adds an object of type E to the collection.
2. void clear( ): Removes all objects from the collection.
3. boolean contains(E elt): Returns true if a specified object is an element within the collection.
4. boolean isEmpty( ): Returns true if the collection has no elements, false otherwise.
5. boolean remove(E elt): Removes a specified object from the collection if it exists.
6. int size( ): Returns the number of elements in the collection.

### Exercise 3: Array Implementation of a Set

Implement a generic class **ArraySet** which implements interface **ISet** using an array to store the elements of the set. You may want to have a look at the **ArrayStack** implementation for inspiration. Remember a set MUST NOT have duplicate values.

### Exercise 4: Linked List Implementation of a Set

Implement a generic class **LinkedListSet** which implements interface **ISet** using a linked list to store the elements of the set. You may want to have a look at the LinkStackInteger implementation. Note, you can create a non public class (omit **public** before the class declaration) **Node** in the same file as the class **LinkedListSet**.