Jemma Tiongson
Comp 182
Prof Wang
Lab8 - Ch.5

1. Read the following articles:

http://www.journaldev.com/1663/java-generics-example-method-class-interface

http://www.javatpoint.com/generics-in-java

http://www.tutorialspoint.com/java/java_generics.htm

https://docs.oracle.com/javase/8/docs/api/java/util/Iterator.html

https://docs.oracle.com/javase/8/docs/api/java/util/ListIterator.html

https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html

2. Copy the code in pages 291 and 292, and make sure that how these codes work.

3. Copy the code in http://www.tutorialspoint.com/java/java_generics.htm, and make sure that how these codes work.

4, Copy the code in page 294, and make sure that how these codes work.

5. Using what we learned in class (such as JCF. implement codes which satisfy the following requirements:

1) Declare a linked list

2) Add five string elements to the linked list.

3) Display linked list contents

4) Add First and Last Element

5) Using linked list get and set method, show the first element, change the first element, and show the updated first element

6) Remove the first and the last element

7) Using add and remove methods add a new element to at the first position and then remove the third position of the linked list.

8) Display the final content of the linked list.

For the completed list of list operations refer to

- https://docs.oracle.com/javase/7/docs/api/java/util/AbstractList.html

- https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html

- https://docs.oracle.com/javase/7/docs/api/java/util/List.html

---

2. Generics Working Code & Results:

```
package com.comp182.pt2;

/**
* Created by JemmaMarie on 6/14/17.
*/
public class MyClass <E> {
  private E theData;
  private int n;

  public MyClass(){
     n = 0;
  } // end constructor

  public MyClass (E initData, int num){
     n = num;
     theData = initData;
  }// end constructor

  public void setData(E newData){
     theData = newData;
  } //end set Data

  public E getData(){
     return theData;
  }

  public int getNum(){
     return n;
  } //end getNum
}// end MyClass

package com.comp182.pt2;
```

```
/**
* Created by JemmaMarie on 6/14/17.
*/
public class Main{
  static public void main(String [] args) {
     MyClass<String> a = new MyClass<String>();
     Double d = new Double(6.4);
     MyClass<Double> b = new MyClass <Double>(d, 51);

     a.setData("Sarah");
     System.out.println(a.getData() + ", "+b.getData());
     System.out.println(a.getNum() +", "+b.getNum());
  }
}
```

Results:
Sarah, 6.4
0, 51

3. GenericStackDemo Source Code & Results:

```
package com.comp182.pt3;

/**
* Created by JemmaMarie on 6/14/17.
*/
public interface GenericStack<E> {
  void push(E v);
  Object pop();
  Object peek();
  boolean isEmpty();
}
package com.comp182.pt3;

import java.util.EmptyStackException;

/**
* Created by JemmaMarie on 6/14/17.
*/
public class LinkedListStack<E> implements GenericStack<E> {
```

```
public LinkedListStack(){
   top = null;
}
private static class Element<E> {
   private E data;
   private Element<E> next;
   Element(E data, Element<E> next){
      this.data = data;
      this.next = next;
   }
}
private Element <E> top;
@Override
public void push(E v){
   if(v == null){
      throw new NullPointerException();
   }
   top = new Element<E>(v, top);

}
@Override
public Object pop(){
   if(isEmpty()){
      throw new EmptyStackException();
   }
   Object data = top.data;
   top = top.next;
   return data;
}
@Override
public Object peek(){
   if(isEmpty()){
      throw new EmptyStackException();
   }
   Object data = top.data;
   top = top.next;
   return data;
}
public boolean isEmpty(){
```

```
    return(top == null);
  }
}
package com.comp182.pt3;

/**
* Created by JemmaMarie on 6/14/17.
*/
public class LinkedListStackDemo {
  public static void main(String [] args){
    GenericStack<Integer> stack = new LinkedListStack<Integer>();
      stack.push(4);
      stack.push(45);
      stack.push(34);
      stack.push(134);
      stack.push(434);
      stack.push(1634);

      while(!stack.isEmpty()){
        System.out.println(stack.pop());
      }

  }
}
```

Result:

```
1634
434
134
34
45
4
```

4. TestLinkedList Demo & Result

```
package com.comp182.pt4;
import java.util.LinkedList;
import java.util.Iterator;

/**
```

```
* Created by JemmaMarie on 6/15/17.
*/
public class TestLinkedList {
  static public void main (String [] args){
     LinkedList<Integer> myList = new LinkedList<Integer>();

     Iterator iter = myList.iterator();
     if(!iter.hasNext()){
        System.out.println("The list is empty");
     }
     for(int i = 1; i <= 5; i++){
        myList.add(new Integer(i));
     }
        iter = myList.iterator();
     while(iter.hasNext()){
        System.out.println(iter.next());
     }
  }
}
```

Result:

```
The list is empty
1
2
3
4
5
```

5. Own Linked List JCF demo & Result

```
package com.comp182.pt5;
import java.util.Iterator;
import java.util.LinkedList;

/**
* Created by JemmaMarie on 6/15/17.
*/
public class SomeList {
  static public void main(String [] args){
```

```
//declared a linked list
LinkedList<String> jemmasList = new LinkedList<String>();

//added 5 string elements
jemmasList.add("red");
jemmasList.add("green");
jemmasList.add("blue");
jemmasList.add("indigo");
jemmasList.add("violet");

//display list
System.out.println("There are these many colors of the rainbow: " + jemmasList.size());
System.out.println("The colors of the rainbow are: ");
Iterator iter = jemmasList.listIterator();
while(iter.hasNext()){
   System.out.print(iter.next()+" ");
}

//add first and last element
System.out.println("\nLet's add more colors!");
jemmasList.addFirst("pink");
jemmasList.addLast("burgundy");

//get, set, and show new element
System.out.println("Get first color on my list: " +jemmasList.get(0));
jemmasList.set(0,"magenta");
System.out.println("New first color on my list: " +jemmasList.get(0));

//removed first and last elements
System.out.println("Removing first and last colors on the list");
jemmasList.removeFirst();
jemmasList.removeLast();

//added new element
System.out.println("Add new first color on list");
jemmasList.addFirst("teal");

System.out.println("Removing the 3rd color on the list");
jemmasList.remove(2);
```

```
        //displaying final list
        System.out.println("Jemma's final color list: ");
        iter = jemmasList.listIterator();
        while(iter.hasNext()){
            System.out.print(iter.next()+" ");
        }
    }
}
```

Result:
There are these many colors of the rainbow: 5
The colors of the rainbow are:
red green blue indigo violet
Let's add more colors!
Get first color on my list: pink
New first color on my list: magenta
Removing first and last colors on the list
Add new first color on list
Removing the 3rd color on the list
Jemma's final color list:
teal red blue indigo violet