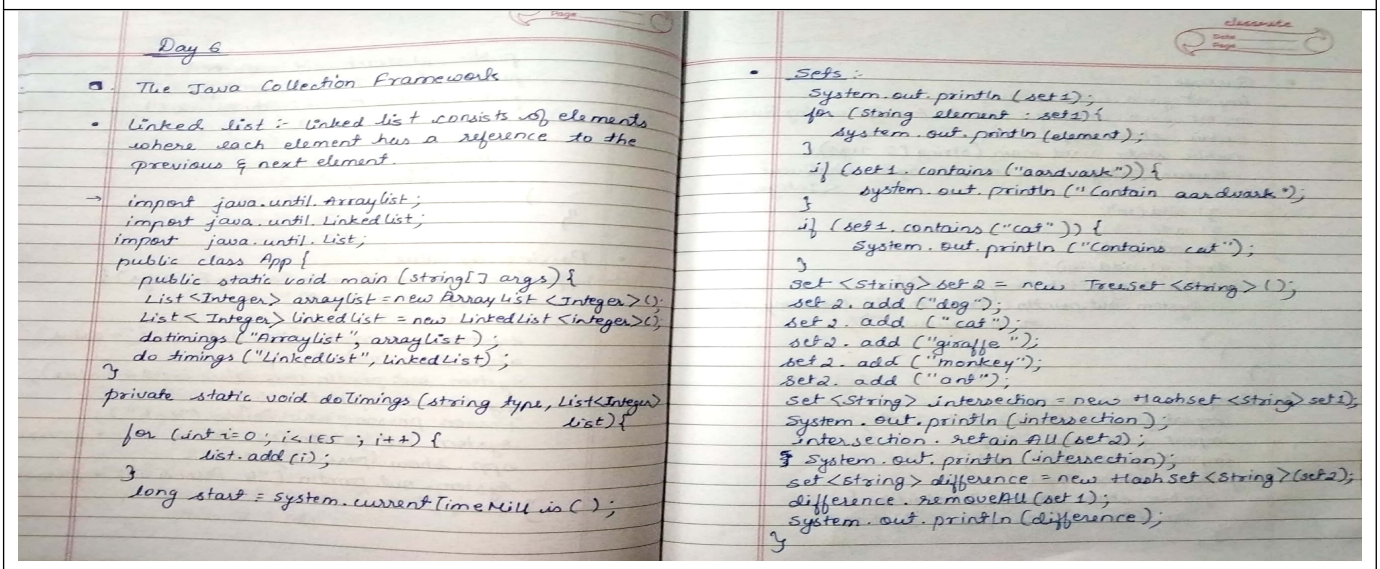
DAILY ASSESSMENT FORMAT

Date:	13-06-2020	Name:	Yamunashree N
Course:	Java	USN:	4AL17EC097
Topic:	The java collection framework Complex data structures Debugging in eclipse.	Semester & Section:	6TH SEM 'B' SEC

AFTERNOON SESSION DETAILS



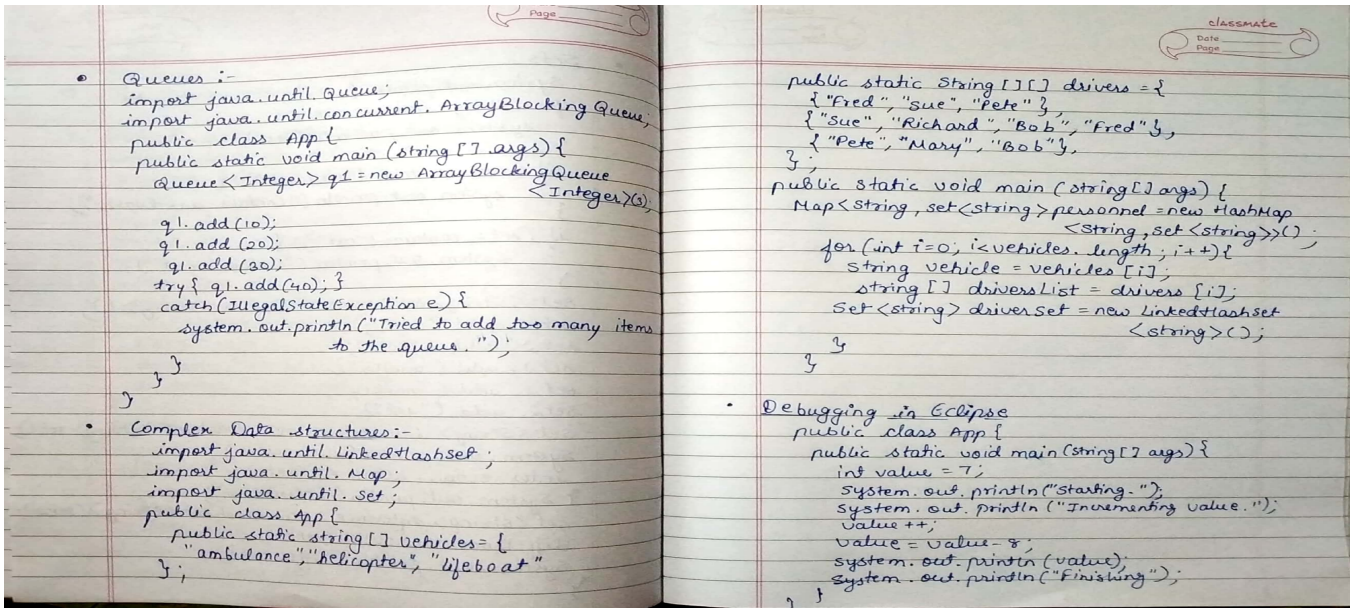
Day 6

- The Java Collection Framework
- Linked list :- Linked list consists of elements where each element has a reference to the previous & next element.

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
public class App {
  public static void main (String[] args) {
    List<Integer> arraylist = new ArrayList<Integer>();
    List<Integer> linkedList = new LinkedList<Integer>();
    doTimings ("ArrayList", arrayList);
    doTimings ("LinkedList", LinkedList);
  }
  private static void doTimings (String type, List<Integer>
                                                  list) {
    for (int i=0; i<=15; i++) {
      list.add(i);
    }
    long start = System.currentTimeMillis();
```

- Sets :

```
System.out.println (set1);
for (String element : set1) {
  System.out.println (element);
}
if (set1.contains ("aardvark")) {
  System.out.println ("Contain aardvark");
}
if (set1.contains ("cat")) {
  System.out.println ("Contains cat");
}
Set<String> set2 = new TreeSet<String>();
set2.add ("dog");
set2.add ("cat");
set2.add ("giraffe");
set2.add ("monkey");
set2.add ("ant");
Set<String> intersection = new HashSet<String>(set1);
System.out.println (intersection);
intersection.retainAll (set2);
System.out.println (intersection);
Set<String> difference = new HashSet<String>(set2);
difference.removeAll (set1);
System.out.println (difference);
}
```



- Queues :-

```
import java.util.Queue;
import java.util.concurrent.ArrayBlockingQueue;
public class App {
  public static void main (String[] args) {
    Queue<Integer> q1 = new ArrayBlockingQueue
                                    <Integer>(5);
    q1.add(10);
    q1.add(20);
    q1.add(30);
    try { q1.add(40); }
    catch (IllegalStateException e) {
      System.out.println ("Tried to add too many items
                          to the queue.");
    }
  }
}
```

- Complex Data structures:-

```
import java.util.LinkedHashSet;
import java.util.Map;
import java.util.Set;
public class App {
  public static String[] vehicles = {
    "ambulance", "helicopter", "lifeboat"
  };
  public static String[][] drivers = {
    {"Fred", "Sue", "Pete"},
    {"Sue", "Richard", "Bob", "Fred"},
    {"Pete", "Mary", "Bob"},
  };
  public static void main (String[] args) {
    Map<String, Set<String>> personnel = new HashMap
                                <String, Set<String>>();
    for (int i=0; i<vehicles.length; i++) {
      String vehicle = vehicles[i];
      String[] driversList = drivers[i];
      Set<String> driverSet = new LinkedHashSet
                                    <String>();
    }
  }
```

- Debugging in Eclipse

```
public class App {
  public static void main (String[] args) {
    int value = 7;
    System.out.println ("Starting.");
    System.out.println ("Incrementing value.");
    value++;
    value = value - 8;
    System.out.println (value);
    System.out.println ("Finishing");
  }
}
```