UNIVERSITY OF GLASGOW – SCHOOL OF COMPUTING SCIENCE
CSC 1009 OBJECT-ORIENTED PROGRAMMING

Programming Laboratory Wk06 Report
Name: Edward foo Seow Wei
Student ID: 2101258 (SIT ID)
Github ID: Anermers95

Important:

1. Export/Save as/Print as PDF, including rename the file into the following format, small letters, without space: sit-student-id_lab_wkXX.pdf
2. For problem question, type your answer
3. For programming question, take several screen capture on the outputs as an evidences.

ANSWERS:

Question 1:

Week6linkedlist class:

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.LinkedList;

public class Week6LinkedList {


    Week6LinkedList()
    {

    }

    void addAndSort(LinkedList<Integer> linkedList, int value)
    {

        System.out.println(linkedList);
        linkedList.add(value);

        Collections.sort(linkedList);
        System.out.println(linkedList);
    }

    void Swap(LinkedList<Integer> linkedList, int indexOne, int indexTwo)
    {
```

```
        Collections.swap(linkedList, indexOne, indexTwo);
        System.out.println(linkedList);
    }

    void findValue(LinkedList<Integer> linkedList, int searchVal)
    {
        ArrayList<Integer> temp = new ArrayList<>();

        for(int i = 0; i < linkedList.size(); ++i)
        {
            if(linkedList.get(i) == searchVal)
                temp.add(i);
        }

        if(temp.isEmpty())
        {
            System.out.println(-1);
        }
        else
        {
            System.out.println("The element is found in index(s): "+ temp);
        }
    }



}
```

Output:

```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

t.java\jdt_ws\week06_d4eef4aa\bin' 'Main'
[1, 3, 5, 7, 9, 11]
[1, 1, 3, 5, 7, 9, 11]
[11, 1, 3, 5, 7, 9, 1]
-1
```

Question 2:

Week6HashMap class:

```
import java.util.HashMap;

public class Week6HashMap {

    Week6HashMap()
    {

    }


    void addAndSort (HashMap<Integer,Integer> hashmap, int value)
    {
       //Store the index at where to insert the value
       int indexVal = 0;
       //Iterate through the list to find the value
       for(int i = 0; i < hashmap.size(); ++i)
       {
           //If value in list is lower than value skip to next
           if(hashmap.get(i) < value)
           {
               indexVal = i+1;
               continue;
           }
           //Found value store index and break loop
           indexVal = i;
           break;
       }
       //Insert into list
       hashmap.put(indexVal, value);
        System.out.println(hashmap);
    }

    void swapValues(HashMap<Integer,Integer> hashmap, int value, int
secondValue)
    {
        int x = hashmap.get(value);
        int y = hashmap.get(secondValue);

        hashmap.replace(value,y);
        hashmap.replace(secondValue, x);

        System.out.println(hashmap);
    }

    void findValue(HashMap<Integer,Integer> hashmap, int value)
```

```
    {
        if(hashmap.containsValue(value))
        {
            for(int i : hashmap.values())
            {
                if(i == value)
                {
                    System.out.println("Value is at key: "+ hashmap.keySet());
                    break;
                }
            }
        }
        else
            System.out.println(-1);
    }
}
```

Output:

```
{0=1, 1=3, 2=5, 3=6, 4=9, 5=11}
{0=11, 1=3, 2=5, 3=6, 4=9, 5=1}
-1
PS D:\sit\CSC10090OP\week06>
```

For these 3 questions, A hashmap is not need because the order when inserted is not maintained and also if a duplicate value is detected, it will simply replace it hence not allowing for duplicate values.