Report on s1709906 's revision attempt at Inf1OP Mock Programming Exam

Generated by Automarker on Fri May 4 21:30:27 BST 2018

Question 1

Part 1all

Compiling Question1.java with the basic tests given to students in the exam succeeded.

Passed all 4 basic tests.

Compiling the submitted Question1.java with the test file worked fine.

There were 2 failures out of the 21 tests.

The tests that failed were:

testMain

testMainTheirs

Problem	Key evidence	Effect on mark
1d) Error in main, but contains right method calls	main failed, SA	-5 Total: -5

Marks for this part: 45 / 50

Submitted Question1.java

```
1  import java.util.ArrayList;
2  import java.util.Arrays;
3
4  public class Question1 {
5
6   public static ArrayList<Integer> cayley() {
7    ArrayList<Integer> result = new ArrayList<Integer>();
8    for (int i = 1; i <= 9; i++) {
9     for (int j = 0; j <= 9; j++) {
```

```
    if (i*10 + j == i*j + i+j) {
     result.add(i*10 + j);
    }
   }
  }

//  System.out.println(Arrays.toString(result.toArray()));
  return result;
 }

 public static boolean isVowel(char c) {
  boolean a = c == 'a' || c == 'A';
  boolean e = c == 'e' || c == 'E';
  boolean i = c == 'i' || c == 'I';
  boolean o = c == 'o' || c == 'O';
  boolean u = c == 'u' || c == 'U';

  return a || e || i || o || u;

 }
 public static int[] countVowels(String[] strings) {
  int[] result = new int[strings.length];

  for (int i = 0; i < strings.length; i++) {
   for (int j = 0; j < strings[i].length(); j++) {
    if (isVowel(strings[i].charAt(j))) {
     result[i]++;
    }
   }
  }

 // System.out.println(Arrays.toString(result));
  return result;
 }

 public static int countVowels2(String s) {
  int count = 0;
  for (int i = 0; i < s.length(); i++) {
   if (isVowel(s.charAt(i))) {
    count++;
   }
  }
  return count;
 }

 public static String mostVowels(String[] strings) {
  if (strings == null) {
   return "";
  }

```

```
  String result = "";
  int max = 0;
  for (String s : strings) {
   if (countVowels2(s) > max) {
    max = countVowels2(s);
    result = s;
   }
  }

//  System.out.println(result);
  return result;
 }


 public static void main(String[] args) {
  System.out.println(Arrays.toString(cayley().toArray()));
  String[] a = new String[args.length];
 // String[] b = {"HELLO", "world"};
  for (int i = 0; i < args.length; i++) {
   a[i] = args[i];
  }

 // System.out.println(Arrays.toString(a));
  System.out.println(Arrays.toString(countVowels(a)));
  System.out.println(mostVowels(a));


 }
}
```

Marks for Question 1: 45/50

Question 2

Part 2all

Compiling AdvancedTutorial.java with the basic tests given to students in the exam succeeded.

Passed all 4 basic tests.

Compiling the submitted AdvancedTutorial.java with the test file worked fine.

There were 1 failures out of the 15 tests.

The tests that failed were:

testMain0

Problem	Key evidence	Effect on mark
2d) main doesn't handle wrong arguments properly	main0	-3 Total: -3

Marks for this part: 47 / 50

Submitted AdvancedTutorial.java

```
1  import java.util.ArrayList;
2  import java.util.Arrays;
3  import java.util.HashMap;
4
5  class AdvancedTutorial extends Tutorial {
6
7   private String topic;
8
9   public AdvancedTutorial(String topic, String tutor, String
    time) {
10   super(tutor, time);
11   this.topic = topic;
12  }
13
14  public String toString() {
15   String s = "Topic:␣" + topic + "␣" + super.toString();
16   return s;
17  }
18
19  public boolean couldCombineWith(AdvancedTutorial at) {
20   boolean a = this.getTime().equals(at.getTime());
21   boolean b = this.topic.equals(at.topic);
22   return a && b;
23  }
24
25 // public static HashMap<String, AdvancedTutorial> construct
    (String[] strings) {
26 //  ArrayList<AdvancedTutorial> arraylist = new ArrayList<
    AdvancedTutorial>();
27 //  HashMap<String, AdvancedTutorial> hashmap = new HashMap<
    String, AdvancedTutorial>();
28 //
29 //  ArrayList<String> keys = new ArrayList<String>();
30 //
31 //  for (int i = 0; i < strings.length; i++) {
32 //   if (i % 3 == 0) {
33 //     AdvancedTutorial at = new AdvancedTutorial(strings[i
    ], strings[i+1], strings[i+2]);
34 //     arraylist.add(at);
```

```
//      hashmap.put(strings[i],at);
//    }
//  }
//
////  for (int i = 0; i < strings.length; i++) {
////   if (i%3 == 0) {
////    keys.add(strings[i]);
////    for (String key : keys) {
////     hashmap.put(key, arraylist);
////    }
////   }
////  }
//
// // System.out.println(Arrays.toString(arraylist.toArray()
 ));
//  return hashmap;
// }

 public String getTopic() {
  return topic;
 }

 public static HashMap<String, ArrayList<AdvancedTutorial>>
 construct(String[] data) {
  HashMap<String, ArrayList<AdvancedTutorial>> sorted = new
 HashMap<String, ArrayList<AdvancedTutorial>>();
  ArrayList<AdvancedTutorial> tuts = new ArrayList<
 AdvancedTutorial>();

  for (int i = 0; i < data.length; i+=3) {
   AdvancedTutorial a = new AdvancedTutorial(data[i], data[i
 +1], data[i+2]);
   tuts.add(a);
  }

  for (AdvancedTutorial t : tuts) {
   String topic = t.getTopic();
   ArrayList<AdvancedTutorial> sameTopic = new ArrayList<
 AdvancedTutorial>();
   for (AdvancedTutorial t2 : tuts) {
    String topic2 = t2.getTopic();
    if (topic.equals(topic2)) {
     sameTopic.add(t2);
    }
   }
   sorted.put(topic, sameTopic);
   }
  return sorted;
 }

```

```
79
80  public static void main(String[] args) {
81   if (args.length % 3 != 0) {
82    System.out.println("Wrong␣number␣of␣arguments!");
83   }
84
85   String[] strings = new String[args.length];
86   for (int i = 0; i < strings.length; i++) {
87    strings[i] = args[i];
88   }
89
90   System.out.println(construct(strings).toString());
91  }
92 }
```

Marks for Question 2: 47/50

Total marks: 92 / 100