Report on s1709906's revision attempt at 2014's Inf1OP Programming Exam (sitting 2)

Generated by Automarker on Wed May 2 19:20:59 BST 2018

Question 1

Part 1a

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

Passed all 6 basic tests.

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

There were 1 failures out of the 15 tests.

The tests that failed (ordered as by JUnit!) were:

1) testSetter3(TestLength)

Problem	Key evidence	Effect on mark
Error in comparison (treatment of incorrect args)	setter3/4 test failed	-3 Total: -3

Marks for this part: 22 / 25

Submitted Length.java

```
1
2  public class Length {
3
4   private int number;
5   private String units;
6
7   public Length() {
8    number = 0;
9    units = "km";
```

```
10  }
11
12  public int getNumber() {
13   return number;
14  }
15
16  public void setNumber(int newNumber) {
17   if (number >= 0) {
18    this.number = newNumber;
19   }
20  }
21
22  public String getUnits() {
23   return units;
24  }
25
26  public void setUnits(String newUnit) {
27   boolean a = newUnit.equals("km");
28   boolean b = newUnit.equals("miles");
29   boolean c = newUnit.equals("minutes");
30
31   if (a || b || c) {
32    this.units = newUnit;
33   }
34  }
35
36  public void convert(String newUnits, double rate) {
37   setUnits(newUnits);
38   setNumber((int) Math.round(getNumber() / rate));
39  }
40
41  public String toString() {
42   return getNumber() + "␣" + getUnits();
43  }
44
45 // public static void main(String[] args) {
46 //  Length l1 = new Length();
47 //  l1.setNumber(40);
48 //  l1.setUnits("minutes");
49 //  l1.convert("miles", 1.60934);
50 //  System.out.println(l1.toString());
51 // }
52 }
```

Part 1b

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

Passed all 2 basic tests.

Compiling the submitted GymSession.java with the test file worked fine.
Passed all 11 tests.

Marks for this part: 25 / 25

Submitted GymSession.java

```
1  import java.util.HashMap;
2
3  class GymSession extends ExerciseSession {
4
5   private HashMap<String,Length> machines = new HashMap<
   String,Length>();
6
7   public GymSession(String type, String place) {
8    super(type, place);
9    machines = new HashMap<String,Length>(8);
10  }
11
12  public void addMachine(String name, Length use) {
13   machines.put(name, use);
14  }
15
16  public String toString() {
17   String s = super.toString() + "\nMachines:";
18   for (String machine : machines.keySet()) {
19    s = s + "\n" + machine + "␣" + machines.get(machine);
20   }
21   return s;
22  }
23
24  public static void main(String[] args) {
25   GymSession g1 = new GymSession("Gym", "The␣Pleasance");
26   Length l1 = new Length();
27   l1.setNumber(10);
28   l1.setUnits("minutes");
29   g1.addMachine("Treadmill", l1);
30   g1.addMachine("Cross-trainer", l1);
31   System.out.print(g1.toString());
32  }
33
34 }
```

Marks for Question 1: 47/50

Question 2

Part 2a-e

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

Passed all 5 basic tests.

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

Passed all 7 tests.

Marks for this part: 50 / 50

Submitted Inequalities.java

```
1  import java.util.Arrays;
2
3  public class Inequalities {
4
5   public static int dotProduct(int[] a, int[] b) {
6
7    int result = 0;
8
9    if (a.length != b.length) {
10    return 0;
11   }
12
13   for (int i = 0; i < a.length; i++) {
14    int product = a[i] * b[i];
15    result += product;
16   }
17
18   return result;
19  }
20
21  public static int[] concatenate(int[] a, int[] b) {
22   int[] c = new int[a.length + b.length];
23
24   for (int i = 0; i < a.length; i++) {
25    c[i] = a[i];
26   }
27
28   for (int i = 0; i < b.length; i++) {
29    c[i + a.length] = b[i];
30   }
31
32   return c;
```

```
33  }
34
35  public static boolean cs(int[] a, int[] b) {
36   boolean one = a.length == b.length;
37   double lhs = Math.pow(dotProduct(a,b), 2);
38   double rhs = (dotProduct(a,a) * dotProduct(b,b));
39   boolean two = lhs <= rhs;
40
41   return one && two;
42  }
43
44  public static boolean amgm(int[] a) {
45   int sum = 0;
46   int n = a.length;
47
48   for (int i = 0; i < a.length; i++) {
49    sum = sum + a[i];
50   }
51
52   double arithmeticMean = sum / n;
53
54   int product = 0;
55   for (int i = 0; i < a.length; i++) {
56    product = product * a[i];
57   }
58
59   double geometricMean = Math.pow(product, 1/n);
60
61   return arithmeticMean >= geometricMean;
62  }
63
64
65  public static void main(String[] args) {
66
67 //  // test dotProduct
68 //  int[] a = {2,1};
69 //  int[] b = {3,4};
70 //
71 //  System.out.println(dotProduct(a,b));
72 //
73 //  // test concatenate
74 //  int[] c = {5,6};
75 //  int[] d = {1,2,3};
76 //
77 //  System.out.println(Arrays.toString(concatenate(c,d)));
78 //
79 //  // test cs
80 //  System.out.println(cs(a,b));
81 //
82 //  // test amgm
```

```
//  int[] e = {4,9};
//  System.out.println(amgm(e));

  int[] a = new int[(args.length - 1)/2];
  int[] b = new int[(args.length - 1)/2];
  int n = Integer.parseInt(args[0]);

  for (int i = 0; i < n; i++) {
   a[i] = Integer.parseInt(args[i+1]);
  }

  for (int i = 0; i < n; i++ ) {
   b[i] = Integer.parseInt(args[i+1+n]);
  }

  System.out.println("CS held: " + cs(a,b));
  System.out.println("AMGM held: " + amgm(concatenate(a,b)))
 ;
//  System.out.println(Arrays.toString(a));
//  System.out.println(Arrays.toString(b));

 }


}
```

Marks for Question 2: 50/50

Total marks: 97 / 100