Inf1 OOP Main Exam: Sample Solutions

Given the limited nature of the tasks in these questions, it is difficult for students to demonstrate excellence over and above giving correct answers. Therefore, in general a set of answers that produces the correct results will receive full marks.

Answers which fail to produce correct results will still be given partial credit, according to the following guidelines:

- If there appears to be a good understanding of the logic of the task, but there are minor syntactic errors, then 25% will be deducted from the marks available.
- If there is a significant error in the logic of the task, then 50% will be deducted from the marks available.

Obviously, these penalties will be adjusted according to the severity of the problem.

Question 1 Solutions

In some of the questions, and the sample solutions, the variable `N` is used to indicate the length of arrays. Although not standard practice in Java, this follows the convention used in the course textbook (and the lectures).

`OneA.java`

```java
public class OneA {

     public static double harmonicMean(int[] nums) {
         int N = nums.length;
         double count = 0.0;
         for (int i = 0; i < N; i++) {
             count += (1.0 / nums[i]);
         }
         return 1 / count * N;
     }
 }
```

`OneB.java`

```java
public class OneB {

     public static double[] movingAvg(double[] data, int len) {
         int N = data.length;
         int w = len - 1;
         double[] result = new double[N - w];
         for (int i = w; i < N; i++) {
             double sum = 0.0;
             for (int j = i - w; j <= i; j++) {
                 sum += data[j];
             }
             result[i - w] = sum / len;
         }
         return result;
     }
 }
```

OneC.java

This question may look somewhat complex at first sight, but this is mainly due the fact that it has been subdivided into fairly fine-grained tasks. Apart from encouraging students to modularize their code into functions, the granularity makes it easier for students to get partial credit even if they are unable to complete the whole question.

They will have seen several examples of the idiom for swapping items in an array using a temporary variable.

`StdIn.java` is a library that accompanies the course textbook. They have been given exercises involving its use, and Javadoc for the class will be made available online during the exam.

```
import java.util.ArrayList;

 public class OneC {

     public static String[] getDomainLabels(String url) {
         String[] urlparts = url.split("/");
         String[] labels = urlparts[0].split("\\.");
         return labels;
     }

     public static void reverseArray(String[] labels) {
         int N = labels.length;
         for (int i = 0; i < N / 2; i++) {
             String temp = labels[i];
             labels[i] = labels[N - 1 - i];
             labels[N - 1 - i] = temp;
         }
     }

     public static String arrayToString(String[] labels) {
         String result = labels[0];
         for (int i = 1; i < labels.length; i++) {
             result = result + "." + labels[i];
         }
         return result;
     }

     public static ArrayList<String> textToReversedDomains(String filename) {
         StdIn.redirectInput(filename);
         ArrayList<String> reversed = new ArrayList<String>();

         while (!StdIn.isEmpty()) {
             String nl = StdIn.readLine();
             if (nl.startsWith("www")) {

                 String[] labels = getDomainLabels(nl);
                 reverseArray(labels);
                 String result = arrayToString(labels);
                 reversed.add(result);
             }
         }
         return reversed;
     }
 }
```

QuestionOneTester.java

The main rationale for this question is to encourage students to budget some time for testing. They have seen many examples of testing code which runs some methods inside `main()`. If they have two reasonable-looking calls for each of their three methods, they will receive full marks.

Question 1 Supplied Files

The following skeleton and data files are provided to the students.

OneA.java

```
public class OneA {

     public static double harmonicMean(int[] nums) {
         // ADD CODE HERE
     }
 }
```

OneB.java

```
public class OneB {

     public static double[] movingAvg(double[] data, int len) {
         // ADD CODE HERE
     }
 }
```

OneC.java

```
import java.util.ArrayList;

 public class OneC {

     public static String[] getDomainLabels(String url) {
         // ADD CODE HERE
     }

     public static void reverseArray(String[] labels) {
         // ADD CODE HERE
     }

     public static String arrayToString(String[] labels) {
         // ADD CODE HERE
     }

     public static ArrayList<String> textToReversedDomains(String filename) {
         // ADD CODE HERE
     }

 }
```

data.txt

```
Hotel Regina Barcelona
www.reginahotel.com
Restaurant Barcelona
www.TablaoCordobes.com
Flights Barcelona Madrid
Book now with Vueling Airlines. Flying today means Vueling
www.vueling.com
Book Barcelona Hotels
Barcelona Hotels. Save up to 70%. Lowest Prices Guaranteed. ABTA
www.TravelRepublic.co.uk/Barcelona
Cheap Hotels In Barcelona
Low Prices On Hotels In Barcelona Guaranteed Here. Save £ Now!
www.Barcelona.Travel-Betterdeals.com
Barcelona Hotels
Superb Discounts! Save up to 50% Compare Our Great Choice of Hotels
www.Barcelona-Hotel.locomotour.co.uk
Edinburgh Vouchers
1 ridiculously huge Coupon a Day. Get 50-90% Off Edinburgh's Best!
www.GROUPON.co.uk/Edinburgh+Coupons
Hotel Barcelona Expedia
Great range of Barcelona Hotels. Choose your perfect one and save.
www.expedia.co.uk
```

Question 2 Solutions

Although this question expects students to have a basic knowledge of vectors, they have been given several labsheets during the semester which involve similar sorts of mathematical structures, including 2D points, vectors and rectangles.

Point.java

```
public class Point {

     private int x, y;

     public Point(int x, int y) {
         this.x = x;
         this.y = y;
     }

     public Point() {
         this(0, 0);
     }

     public int getX() {
         return x;
     }

     public int getY() {
         return y;
     }

     public Point makeMax(Point p) {
         return new Point(Math.max(x, p.getX()), Math.max(y, p.getY()));
     }

     public Point makeMin(Point p) {
         return new Point(Math.min(x, p.getX()), Math.min(y, p.getY()));
     }
 }
```

Vector.java

```
public class Vector {

     private int xdist, ydist;

     public Vector(int xdist, int ydist) {
         this.xdist = xdist;
         this.ydist = ydist;
     }

     public int getXDist() {
         return xdist;
     }

     public int getYDist() {
         return ydist;
     }

     public double magnitude() {
         return Math.sqrt(xdist * xdist + ydist * ydist);
     }

     public Point translate(Point p) {
         return new Point(p.getX() + xdist, p.getY() + ydist);
     }
 }
```

Polygon.java

The low mark available for question (v) is intended not to reflect its difficulty, but rather to ensure that students who run out of time before completing it don't lose too many marks.

```
public class Polygon {

     private Vector[] vectors;

     public Polygon(Vector[] vectors) {
         this.vectors = vectors;
     }

     public Vector[] getVectors(){
         return vectors;
     }

     public double perimeter() {
         double totalLength = 0.0;
         for (Vector v : vectors) {
             totalLength += v.magnitude();
         }
         return totalLength;
     }

     public boolean isClosed() {
         Point current = new Point();
```

```
        for (Vector v : vectors) {
            current = v.translate(current);
        }
        return current.getX() == 0 && current.getY() == 0;
    }

    public double area() {
        if (isClosed()) {
            double area = 0.0;
            Point start = new Point();

            for (Vector v : vectors) {
                Point end = v.translate(start);
                area += start.getX() * end.getY() - end.getX() * start.getY();
                start = end;
            }
            area /= 2;
            return Math.abs(area);

        } else {
            return 0.0;
        }
    }

    public Point[] boundingBox() {
        Point current = new Point();
        Point[] limits = new Point[2];
        limits[0] = new Point();
        limits[1] = new Point();

        for (Vector v : vectors) {
            current = v.translate(current);
            limits[0] = limits[0].makeMin(current);
            limits[1] = limits[1].makeMax(current);
        }
        return limits;
    }
}
```

Question 2 Supplied files

PolygonTester.java

```
public class PolygonTester {

        public static void drawPolygon(Polygon poly) {

                Vector[] vs = poly.getVectors();
                Point[] bbox = poly.boundingBox();

                int width = bbox[1].getX() - bbox[0].getX();
                int height = bbox[1].getY() - bbox[0].getY();

                StdDraw.setCanvasSize(width * 100, height * 100);
                StdDraw.setXscale(bbox[0].getX(), bbox[1].getX());
                StdDraw.setYscale(bbox[0].getY(), bbox[1].getY());

                StdDraw.setPenColor(StdDraw.LIGHT_GRAY);
                StdDraw.rectangle(bbox[0].getX() + width / 2.0, bbox[0].getY() + height
                                / 2.0, width / 2.0, height / 2.0);

                StdDraw.setPenColor();

                Point p = new Point();

                for (Vector v : vs) {
                        Point q = v.translate(p);
                        StdDraw.line(p.getX(), p.getY(), q.getX(), q.getY());
                        p = q;
                }

        }

        public static String bboxstring(Point[] p) {
                return "(" + p[0].getX() + ", " + p[0].getY() + ") x (" + p[1].getX()
                                + ", " + p[1].getY() + ")";
        }

        public static void main(String[] args) {

                Vector v1 = new Vector(2, 2);
                Vector v2 = new Vector(3, -4);
                Vector v3 = new Vector(-5, 2);

                Vector[] vs = { v1, v2, v3 };

                Polygon p = new Polygon(vs);

         System.out.printf("Method: perimeter\n\tExpected: 13.21\n\tActual: %.2f\n",
                 p.perimeter());

         System.out.println("Method: isClosed\n\tExpected: true\n\tActual: "
                 + p.isClosed());
```

```
         System.out.printf("Method: area\n\tExpected: 7.00\n\tActual: %.2f\n",
                 p.area(), p.area());

         System.out.println("Method: boundingBox\n\tExpected: (0, -2) x (5, 2)\n\tActual: "
                + bboxstring(p.boundingBox()));
               drawPolygon(p);
       }
}
```