CSCI 131, Exam3 – Review Partial Solutions

1) Type in to a program to get results.

2a)
```
public class Country {
  private String name;   private double pop;   private double literacy;
  public Country(String n, double p, double l) {
    name = n;     pop = p;    literacy = l;  }
  public double getRate() { return (literacy/pop)*100; }
  public String toString() { return name; } }
```

3) Type in to a program to get results.

4a) Type in to a program to get results.

4b) negative integers

5a) `name,id,gpa`; 2 constructors

5b)
```
Student stu = new Student("Francis", 123456);   stu.changeGPA(3.95);
```

5c)
```
public static double computeAverageGPA (Student[] list) {
     double sum = 0;
     for (int i = 0; i < list.length; i++)
         sum = sum + list[i].getGPA();
     return (sum / list.length);   }
```

5d)
```
double avg = computeAverageGPA(seniors);
   for (int i = 0; i < seniors.length; i++)
      if (seniors[i].getGPA() > avg)
         StdOut.println (seniors[i].toString());  // seniors[i].print();
```

5e)
```
public String toString() {
    return String.format("Student named %s, with gpa %.2f and ID %d", name,id,gpa); }
```

7b)
```
double a, b;
    StdOut.print ("Enter two x, y pairs: ");
    a = StdIn.readDouble();  b = StdIn.readDouble();
    Point p1 = new Point(a, b);
    a = StdIn.readDouble();  b = StdIn.readDouble();
    Point p2 = new Point(a, b);
    if (p1.distanceFromOrigin() < p2.distanceFromOrigin())
       StdOut.println (p2 + " is further from the origin.");
    else if (p2.distanceFromOrigin() < p1.distanceFromOrigin())
       StdOut.println (p1 + " is further from the origin.");
    else
       StdOut.println (p1 + " and " + p2 + " are the same distance from the origin.");
```

7d)
```
public void move(double mx, double my) { x = x + mx; y = y + my; }
```

7e)
```
public void moveTowards(Point T) { x = (x + T.x)/2;  y = (y + T.y)/2;  }
```

7f)
```
public static Point centerOfGravity (Point[] all) {
     double sumX=0;  double sumY = 0;
     for (int p = 0; p < all.length; p++) {
         sumX = sumX + all[p].getX();   sumY = sumY + all[p].getY();  }
     return new Point(sumX/all.length, sumY/all.length);  }
```

13d) `=` is assignment

`==` is testing equality for primitive types

`.equals()` is equality test for some class types (e.g. String)

14) Need to add the missing base case

15)
```
public static boolean containsDuplicates(double[] vals) {
     for (int i = 0; i < vals.length; i++ )
        if( contains(vals, i+1, vals.length, vals[i] ))
            return true;
    return false;  }
```

16a) lines 3&4 are the base case; 16b) lines 6-11 are the general case;

17a) Type in to a program to get results.

17b) Yes.

18)
```
public static int countMatches( String  a, String b) {
     int ct = 0;  int min = a.length();
     if (b.length() < min)   min = b.length();
     for (int i = 0; i < min; i++)
          if (b.charAt(i) == a.charAt(i))  ct++;
     return ct; }
```
20)
```
public static void printTitleCase( String  s) {
     if (s.length() < 1) return;
     StdOut.print (Character.toUpperCase(s.charAt(0)));
     for (int i = 1; i < s.length(); i++)
        if ((s.charAt(i-1) == ' '))
              StdOut.print(Character.toUpperCase(s.charAt(i)));
        else  StdOut.print(s.charAt(i));
     StdOut.println();  }
```
22)
```
double [][] grid = new double [5][8];
   for (int r = 0; r < 5; r++)
       for (int c = 0; c < 8; c++)   grid[r][c] = 17.0;
   for (int r = 0; r < 5; r++) {
       for (int c = 0; c < 8; c++)   StdOut.printf("%3.1f ", grid[r][c]);
       StdOut.println();
   }
```
23)
```
for (int r = 0; r < rows; r++)            {
      result[r] = 0;
      for (int c = 0; c < cols; c++)
          result[r] += (V[c] * M[r][c]);      }
```
25)
```
public static int countPositive (int [] list, int n) {
     int ct = 0;
     for (int i = 0; i < n; i++)
         if (list[i] > 0)  // (list[i] % 2 == 0).  ct++;
     return ct;  }
```
28)
```
for (int t = 0; t < T; t++) {
      int cash = stake;
      while (cash > 0 && cash < goal) {
         bets++;
         if (Math.random() < 0.5)  cash++;
         else    cash--;
      }
      if (cash == goal)    wins++;
   }
```
31) The output is going to the console.

32)
```
public static int most (int [] list) {
    int max = 0;
    for (int i = 0; i < list.length; i++)
        if (list[i] > list[max])   max = i;
    return max;  }
```
36a)
```
public static int firstVowel (String word) {
    for (int i = 0; i < word.length(); i++) {
        char f = word.charAt(i);
        if (f == 'a' || f == 'e' || f == 'i' || f == 'o' || f == 'u')
            return i;
    }
    return -1;  }
```
37) Type into a program to get results

38) `first = login.charAt(0); middle = login.charAt(1); last = login.substring(2,6);`

String Problems) // Returns first half of a string

```
public static String half(String s) { int n = s.length(); return s.substring(0, n/2);
```