Practice designing classes with constructors, instance fields, and instance methods.

1. **Tracing programs**
   
a. What gets printed to the screen when we execute the MysteryClient class on the left?

   *The first line in main() creates an object of type Mystery, and a reference variable pointing to it called m. It does not actually display anything, but here is what is in memory after that instruction:*

   ![Memory Diagram 1](image1)

   *Then the second line in main() executes, which calls the display() method in the Mystery class using object m. This method displays the concatenation of str and the parameter s, which the calling method has set to " again". So the output is:*

   hello again

   b. What gets printed to the screen when we execute the MysteryClient class on the left?

   *The first line creates an object of type Mystery, and a reference to it called m. This doesn't display anything, but here's what's in memory after that instruction:*

   ![Memory Diagram 2](image2)

   *The next line passes a value of 3 to the parameter z in the display method. The display method tests whether 3 divides 15 and 27 (which is true), so it prints out:*

   divides

   *The next line creates another object of type Mystery, and a reference to it called m2. Here's what that creates in memory:*

   ![Memory Diagram 3](image3)

   *The next line calls the reduce method, and passes a value of 4 to the z parameter. The reduce method sets the x field of m2 to its previous x field divided by z, and then sets the y field to its previous y field divided by z. Here's what m2 looks like in memory afterwards:*

   ![Memory Diagram 4](image4)

   *The last line in main calls the display method, and passes a value of 4 to the z parameter. The display method checks whether z divides m2's x attribute (which is true) and whether z divides m2's y value (which is false). So it displays:*

   too bad
2. Creating your own classes
   a. Create a Point3D class.

```java
public class Point3D {
    // doubles would work too.
    // Since the 2d Point class uses ints, I'll stick with ints.
    public int x = 0;
    public int y = 0;
    public int z = 0;

    public Point3D(int x, int y, int z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }

    public void translate(int dx, int dy, int dz) {
        x += dx;
        y += dy;
        z += dz;
    }

    public String toString() {
        return "(" + x + ", " + y + ", " + z + ")";
    }

    public static void main(String[] args) {
        Point3D p1 = new Point3D(6, 5, 9);
        Point3D p2 = new Point3D(-5, 7, -2);

        System.out.println("p1 = " + p1);
        System.out.println("p2 = " + p2);

        p1.translate(1, -3, -5);
        System.out.println("translated p1 = " + p1);
    }
}
```
b. Create a Fraction class.

```java
public class Fraction {
    public int numer = 0;
    public int denom = 1;

    public Fraction(int numerator, int denominator) {
        if (denominator < 1)
            throw new IllegalArgumentException(
                "Fraction with denominator < 1");
        numer = numerator;
        denom = denominator;
    }

    public String toString() {
        return numer + " / " + denom;
    }

    public int greatestCommonDivisor() {
        int maxDivisor = 1;
        for (int i = 1; i < numer && i < denom; i++)
            if (numer % i == 0 && denom % i == 0) {
                maxDivisor = i;
            }
        return maxDivisor;
    }

    public void reduce() {
        int gcd = greatestCommonDivisor();
        x = x / gcd;
        y = y / gcd;
    }

    public static void main(String [] args) {
        Fraction f1 = new Fraction(18, 15);
        // the following is equivalent to (shorthand for)
        // System.out.println("f1 = " = f1.toString());
        System.out.println("f1 = " + f1.toString());
        System.out.println("f1's gcd =" + f1.greatestCommonDivisor());
        f1.reduce();
        System.out.println("reduced f1 = " + f1);
        
        Fraction f2 = new Fraction(-10, 35);
        System.out.println("f2 = " + f2);
        System.out.println("f2's gcd =" + f2.greatestCommonDivisor());
        f2.reduce();
        System.out.println("reduced f2 = " + f2);
        
        // This one should cause our program to crash!
        Fraction f3 = new Fraction(2, -5);
    }
}
```