More practice designing classes with constructors, instance fields, and instance methods.

1. Tracing programs

a. What are the values of the variables in main() at POINT 1?

```java
public class MysteryClient
{
    public static void main(String [] args)
    {
        Mystery m = new Mystery("hello");
        // POINT 1
    }
}
```

```java
public class Mystery
{
    public String str = null;
    public Mystery(String s)
    {
        str = s;
    }
}
```

b. What are the values of the variables in main() at POINT 1? and at POINT 2?

```java
public class MysteryClient
{
    public static void main(String [] args)
    {
        Mystery m1 = new Mystery(15, 9);
        Mystery m2 = new Mystery(16, 4);
        // POINT 1
        m1.update(3);
        m2.update(4);
        // POINT 2
    }
}
```

```java
public class Mystery
{
    public int x = 0;
    public int y = 0;
    public Mystery(int num1, int num2)
    {
        x = num1;
        y = num2;
    }
    public void update(int z)
    {
        x = x / z;
        y = y * z;
    }
}
```
c. What are the values of the variables in main() at POINT 1 and at POINT 2?

```java
public class MysteryClient
{
    public static void main(String[] args)
    {
        Mystery m1 = new Mystery();
        // POINT 1
        int[] arr1 = m1.getArr(4, 5);
        int[] arr2 = m1.getArr(2, 2);
        // POINT 2
    }
}
```

```java
public class Mystery
{
    public Random rand = new Random();
    public int[] getArr(int len, int max)
    {
        int[] ret = new int[len];
        for(int i=0; i<len; i++)
            ret[i] = rand.nextInt(max);
        return ret;
    }
}
```

d. What are the values of the variables in main() at POINT 1 and at POINT 2 and 3?

```java
public class MysteryClient
{
    public static void main(String[] args)
    {
        Mystery m1 = new Mystery();
        Mystery m2 = new Mystery();
        // POINT 1
        m1.setArr(3, 7);
        m2.setArr(2, 9);
        // POINT 2
        int x = m1.getVal(2);
        int y = m1.getVal(2);
        int z = m2.getVal(0);
        // POINT 3
    }
}
```

```java
public class Mystery
{
    public int[] arr = null;
    public void setArr(int len, int val)
    {
        arr = new int[len];
        for(int i=0; i<len; i++)
            arr[i] = val;
    }
    public int getVal(int pos)
    {
        return arr[pos];
    }
}
```
2. Creating your own classes
Create a Bug class so that the BugClient below will work. This class will represent a little critter that crawls across your screen, left or right. The critter will be represented by the String "X_O>" or "<O_X", depending on which way it is facing. The '_' in the display should be replaced by a number to specify which Bug it is.

Each Bug will need an attribute to specify it's position (how many characters to the right from the beginning of the line), an attribute to specify the direction it is facing, and an attribute to specify an ID number for the bug.

It will also need a constructor, and a method to move it forward in the direction it is facing, a method to turn it to face the opposite direction, a method to get its position, and a toString() method.

```
public class BugClient
{
    public static void display(Bug b)
    {
        int pos = b.getPos();
        for(int i=0; i<pos; i++) {
            System.out.print(" ");
        }
        System.out.println(b.toString());
    }

    public static void main(String [] args)
    {
        Bug b1 = new Bug(1);
        b1.move(10);
        display(b1);
        b1.move(25);
        display(b1);
        b1.turn();
        b1.move(15);
        display(b1);

        Bug b2 = new Bug(8);
        display(b2);
        b2.move(55);
        b2.turn();
        b2.move(13);
        display(b2);
    }
}
```