Rubin Peci
CS 102 Data Structures
Professor Chee Yap
Fall 2019
September 23rd, 2019

1.
Line 1 declares a new Located Rectangle under type Recatangle. Though this is "allowed" by Java, it is not proper convention because the LocatedRectangle class extends the Rectangle class and thus has different methods that they don't all share.

Line 2 will return an error because the top() method is only for LocatedRectangles, and rr was declared as type Rectangle, despite being initialized as a new instance of LocatedRectangle.

Line 3 will use the toString() method of a LocatedRectangle when it is in fact a Rectangle, and though it will run, it will not refer to the "right" toString() method.

2. X is the name of a class because if the program is valid, and there is no line that creates a new instance of a class that is stored in a variable named X, then Y must be a static variable that can be accessed without having to initalize a new instance of a class. Thus, Y is a static variable, making X the name of a class.

3.

```
int[] convert(int n) {
        int[] pair = new int[2];
        if (n <= 31) {                          //jan
                pair[0] = 1;
                pair[1] = n;
        } else if (n <= 59) {           //feb
                pair[0] = 2;
                pair[1] = n - 31;
        } else if (n <= 90) {           //mar
                pair[0] = 3;
                pair[1] = n - 59;
        } else if (n <= 120) {          //apr
                pair[0] = 4;
                pair[1] = n - 90;
        } else if (n <= 151) {          //may
                pair[0] = 5;
                pair[1] = n - 120;
        } else if (n <= 181) {          //june
                pair[0] = 6;
                pair[1] = n - 151;
        } else if (n <= 212) {          //july
                pair[0] = 7;
                pair[1] = n - 181;;
        } else if(n <= 243) {           //aug
                pair[0] = 8;
                pair[1] = n - 212;
        } else if (n <= 273) {          //sep
                pair[0] = 9;
                pair[1] = n - 243;
        } else if (n <= 304) {          //oct
                pair[0] = 10;
                pair[1] = n - 273;
        } else if (n <= 334) {          //nov
                pair[0] = 11;
                pair[1] = n - 304;
        } else if (n <= 365) {          //dec
                pair[0] = 12;
                pair[1] = n - 334;
        }
        return pair;
}
```

4. null, reference, link, address, pointer
Null is the odd one out because the other 4 all refer to something that points to some value in memory. A pointer/reference/link/address are all associated with memory and values in memory. Null means that the object/instance points to nothing, or the null value, in ememory.

5a. The default modifier is different from the other 3 modifiers because it is the modifier chosen by Java when no other one is explicitly stated. The scope of the default modifier is limited to anything in the same package as the class its written in.

5b.
The public modifier can be applied to any member, method, or class.

The protected modifier cannot be applied to classes.

The private modifier cannot be applied to classes either.

The default modifier can be applied to any memmber, method, or class.