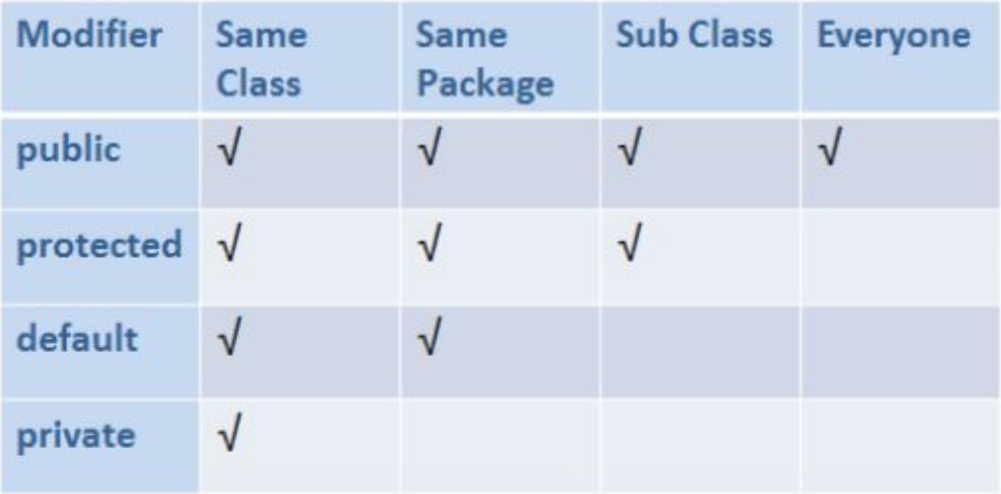
Tutorial 3

Question 2

Access Modifiers



Modifier	Same Class	Same Package	Sub Class	Everyone
public	√	√	√	√
protected	√	√	√	
default	√	√		
private	√			

Question 3

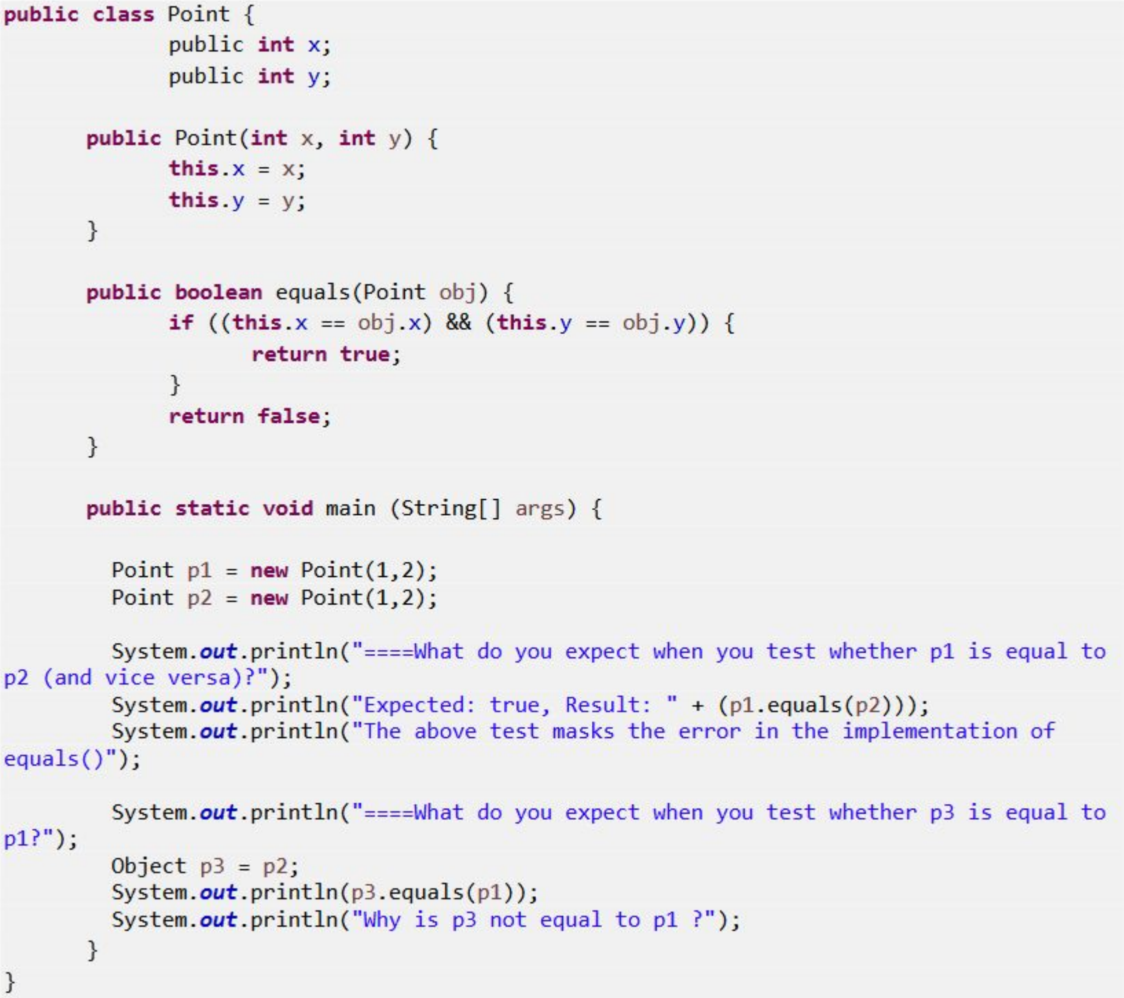
Recap: Method Overriding, Object Class, Equals

- Method overriding - changing implementation of method defined in a parent class
- **Object** class - parent class of all other classes, defines common methods all objects should possess
- **equals()** - method which checks if two objects are equal

equals()

- There exist requirements on how the equals method should be implemented
- Breaking these requirements may break other classes / libraries which expect this ‘contract’ to be upheld

Question 4



```
public class Point {
             public int x;
             public int y;

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

       public boolean equals(Point obj) {
             if ((this.x == obj.x) && (this.y == obj.y)) {
                    return true;
             }
             return false;
       }

       public static void main (String[] args) {

         Point p1 = new Point(1,2);
         Point p2 = new Point(1,2);

         System.out.println("====What do you expect when you test whether p1 is equal to
p2 (and vice versa)?");
         System.out.println("Expected: true, Result: " + (p1.equals(p2)));
         System.out.println("The above test masks the error in the implementation of
equals()");

         System.out.println("====What do you expect when you test whether p3 is equal to
p1?");
         Object p3 = p2;
         System.out.println(p3.equals(p1));
         System.out.println("Why is p3 not equal to p1 ?");
       }
}
```