

Assignment 6 Sample Test Case

Question 1

```
public static void testShape() {  
}
```

Question 2

```
public static void testBubble() {  
    Bubble b = new Bubble(20, 30);  
  
    System.out.println(b.getX() == 20);  
    System.out.println(b.getY() == 30);  
    b.setX(40);  
    System.out.println(b.getX() == 40);  
    System.out.println(b.getY() == 30);  
    b.setY(60);  
    System.out.println(b.getX() == 40);  
    System.out.println(b.getY() == 60);  
  
    // circle fully inside window  
    System.out.println(b.isVisible(100, 100) == true);  
    // circle center on window top edge  
    b.setX(50);  
    b.setY(0);  
    System.out.println(b.isVisible(100, 100) == true);  
    // circle center on window right edge  
    b.setX(99);  
    b.setY(50);  
    System.out.println(b.isVisible(100, 100) == true);  
    // circle center on window bottom edge  
    b.setX(50);  
    b.setY(99);  
    System.out.println(b.isVisible(100, 100) == true);  
    // circle center on window left edge  
    b.setX(0);  
    b.setY(50);  
    System.out.println(b.isVisible(100, 100) == true);  
    // circle center on window top left corner  
    b.setX(0);  
    b.setY(0);  
    System.out.println(b.isVisible(100, 100) == true);  
    // circle center on window top right corner  
    b.setX(99);  
    b.setY(0);  
    System.out.println(b.isVisible(100, 100) == true);  
    // circle center on window bottom right corner  
    b.setX(99);  
    b.setY(99);  
    System.out.println(b.isVisible(100, 100) == true);  
    // circle center on window bottom left corner  
    b.setX(0);  
    b.setY(99);  
    System.out.println(b.isVisible(100, 100) == true);  
}
```

```

// circle bottom edge on window top edge
b.setX(50);
b.setY(-10);
System.out.println(b.isVisible(100, 100) == true);
// circle left edge on window right edge
b.setX(109);
b.setY(50);
System.out.println(b.isVisible(100, 100) == true);
// circle top edge on window bottom edge
b.setX(50);
b.setY(109);
System.out.println(b.isVisible(100, 100) == true);
// circle right edge on window left edge
b.setX(-10);
b.setY(50);
System.out.println(b.isVisible(100, 100) == true);
// circle bottom right edge on window top left corner
b.setX(-7);
b.setY(-7);
System.out.println(b.isVisible(100, 100) == true);
// circle bottom left edge on window top right corner
b.setX(106);
b.setY(-7);
System.out.println(b.isVisible(100, 100) == true);
// circle top left edge on window bottom right corner
b.setX(106);
b.setY(106);
System.out.println(b.isVisible(100, 100) == true);
// circle top right edge on window bottom left corner
b.setX(-7);
b.setY(106);
System.out.println(b.isVisible(100, 100) == true);

// circle bottom edge beyond window top edge
b.setX(50);
b.setY(-11);
System.out.println(b.isVisible(100, 100) == false);
// circle left edge beyond window right edge
b.setX(110);
b.setY(50);
System.out.println(b.isVisible(100, 100) == false);
// circle top edge beyond window bottom edge
b.setX(50);
b.setY(110);
System.out.println(b.isVisible(100, 100) == false);
// circle right edge beyond window left edge
b.setX(-11);
b.setY(50);
System.out.println(b.isVisible(100, 100) == false);
// circle bottom right edge beyond window top left corner
b.setX(-8);
b.setY(-8);
System.out.println(b.isVisible(100, 100) == false);
// circle bottom left edge beyond window top right corner
b.setX(107);
b.setY(-8);
System.out.println(b.isVisible(100, 100) == false);
// circle top left edge beyond window bottom right corner
b.setX(107);
b.setY(107);
System.out.println(b.isVisible(100, 100) == false);
// circle top right edge beyond window bottom left corner
b.setX(-8);

```

```

b.setY(107);
System.out.println(b.isVisible(100, 100) == false);

b.setX(0);
b.setY(0);
System.out.println(b.isIn(2, 2) == true);
System.out.println(b.isIn(10, 0) == true);
System.out.println(b.isIn(0, 10) == true);
System.out.println(b.isIn(7, 7) == true);
System.out.println(b.isIn(11, 0) == false);
System.out.println(b.isIn(0, 11) == false);
System.out.println(b.isIn(8, 8) == false);
}

```

Question 3

```

public static void testModel() {
    Model m = new Model();

    System.out.println(m.getScore() == 0);
    // Testing the bubbles arraylist by directly accessing
    // the arraylist inside the object m. This is not very clean (it only
    // works when the tests are done from inside the class itself) but it
    // allows us to test more methods.
    System.out.println(m.bubbles.size() == 0);
    // Add two bubbles.
    m.addBubble(100, 100);
    m.addBubble(100, 100);
    System.out.println(m.getScore() == 0);
    System.out.println(m.bubbles.size() == 2);
    // The two bubbles we have are somewhere in the square ([0, 99], [0, 99)).
    m.clearInvisibles(200, 200);
    System.out.println(m.getScore() == 0);
    System.out.println(m.bubbles.size() == 2);
    m.clearInvisibles(0, 0);
    System.out.println(m.getScore() == -2);
    System.out.println(m.bubbles.size() == 0);
    // Adding two bubbles again and moving them out of the window.
    m.addBubble(100, 100);
    m.addBubble(100, 100);
    System.out.println(m.getScore() == -2);
    System.out.println(m.bubbles.size() == 2);
    m.moveAll(200, 200);
    System.out.println(m.getScore() == -2);
    System.out.println(m.bubbles.size() == 2);
    m.clearInvisibles(200, 200);
    System.out.println(m.getScore() == -4);
    System.out.println(m.bubbles.size() == 0);
    // Adding two bubbles again and clicking into them to remove them.
    m.addBubble(10, 10);
    m.addBubble(10, 10);
    System.out.println(m.getScore() == -4);
    System.out.println(m.bubbles.size() == 2);
    // The two bubbles we have are somewhere in the square ([0, 9], [0, 9)).
    m.deleteBubblesAtPoint(5, 5);
    System.out.println(m.getScore() == -2);
    System.out.println(m.bubbles.size() == 0);
    // We cannot test the drawAll method because we don't have any graphics object.
    // Adding a bubble again and clearing everything.
    m.addBubble(100, 100);
    System.out.println(m.getScore() == -2);
}

```

```
        System.out.println(m.bubbles.size() == 1);
    }
}
```

Question 4

// This is the final Start Class for your considering, not a test method.

```
public class Start {
    public static void main(String[] args) {
        Shape.testShape();
        Bubble.testBubble();
        Model.testModel();

        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new MyFrame();
            }
        });
    }
}
```