The Need for Specifications

How do you know? (or do you know?)...

- where a DrawingGizmo object is positioned when instantiated?
- how far a DrawingGizmo object moves for each moveForward?
- the precise angle of turn that results from a call to turnClockwise.

Possible “answers” for such questions...

(1) You could trust your instructor.
(2) You could guess, based on the names of the classes and methods.
(3) You could ask the programmer who wrote DrawingGizmo.
(4) You could read the DrawingGizmo code (if you can find it).

OR

(5)

A class specification defines the semantics (behavior) of a class by way of
➢ a class invariant to describe what is always true of the class’s objects.
➢ specifications for each of the classes’ methods.

Each method specification consists of:
• a __________________ (optional),
• a __________________.

states the conditions that are necessary for the method to properly execute.

states what is true when the method completes execution.
Class Specifications

DrawingGizmo Class Specifications

Invariant
A DrawingGizmo object ...
- appears as an arrow within a “Drawing Canvas” window. (This window is 260 pixels wide and 160 pixels high.)
- is either in drawing mode or moving mode.
- in drawing mode the arrow is colored green.
- in moving mode the arrow is colored red.

DrawingTool Class Specifications (continued)

Constructor Method
public DrawingGizmo()

post: a new DrawingGizmo object is created and placed in the center of the Drawing Canvas window. (Note that one DrawingGizmo may cover another.)
and This object is set to drawing mode.
and The arrow for this object is pointing straight up.

Update Methods
public void draw()

post: This object is set to drawing mode.

public void dontDraw()

post: This object is set to moving mode.

public void turnClockwise()

post: This object’s direction of travel is rotated by 30 degrees clockwise from its previous direction.

public void turnCounterclockwise()

post: This object’s direction of travel is rotated by 30 degrees counterclockwise from its previous direction.
public void moveForward()
   pre: A minimum of 20 pixels remain between the arrow and the edge of
       Drawing Canvas in the direction of travel.
   post: This object’s is moved forward by twenty pixels from its previous location.
       and if this object is in drawing mode, then a line segment is drawn across
       the twenty pixel path just traversed. (In moving mode nothing is drawn.)

public void delay2Sec()
   post: All drawing activity for the drawing canvas of this DrawingGizmo
       is suspended for 2 seconds, then resumed.

Given the class specifications for DrawingGizmo, explain the behavior of the
following program.

```java
public class Driver {
    private DrawingGizmo pen, pencil;

    public Driver() {
        pen = new DrawingGizmo();
        pen.moveForward();
        pen.moveForward();
        pencil = new DrawingGizmo();
        pencil.dontDraw();
        pencil.moveForward();
        pencil.moveForward();
        pen.turnClockwise();
        pen.delay2Sec();
        pencil.turnClockwise();
        pencil.turnClockwise();
        pencil.turnClockwise();
        pencil.draw();
        pen.moveForward();
        pencil.moveForward();
    }
}
```