

SECTION D

9 Programming in Java

The following code has been written by a novice Java programmer. You are not required either to understand or to debug the details of how this code draws some particular pattern (a "Dragon").

The programmer finds that the Java compiler complains. Identify any errors and comment on any issues that (even if not strictly invalid Java) are liable to cause problems. You are not required to provide corrections.

```

*****
** Supervison work for June 6th. **
*****
upper class Dragon extends JApplet throws Exception
{
    import javax.swing.*;

    public paint(Graphics g)
    {
        this.g = g;
        drawDragon(DRAWDEPTH, 100, 200, 300, 200);
    }

    /** @title: drawDragon
     *   Function to draw a dragon curve between two points
     *   (x1,y1) and (x2,y2) with depth 'depth'.
     */
    void protected drawDragon(int depth,
                              int x1, int y1, int x2, int y2)
    {
        if (x1 < 0 || x2 < 0)
            if (y1 < 0) raise new Exception("X & Y < 0");
        else assert("Ok so far");
        if (depth == 0) // bottom of recursion
        {
            g.drawLine(x1, y1, x2, y2);
            continue;
        }
        int mpx = (x2 + x1 + y2 - y1) / 2; /* X coord of a new point...
        int mpy = (y2 + y1 - x2 + x1) / 2; ... and Y coord. */
        printf("DEBUG: x= drawDragon(depth-1, mpx, mpy, x1, y1);
        drawDragon(depth+1, mpx, mpy, x2, y2);
    }

    static secret int DRAWDEPTH = 15,
    Graphics g;
}

```

too late for
supern on work!

needs import of java.awt.*

OK.

throws...

y2 not checked

dangle

bool

final?

use
lines not 1)

static conflicts with this.g?

[20 marks]

plq 9
2005

To mark this I start by working through each script, marking ~~each~~ comment as \checkmark , \times or nothing. \checkmark is for a bug detected & explained at least reasonably well or an "ugly" noted. \times is for a comment that is reasonably wrong. For things that are an observation but do not score either $+$ or $-$ I just leave a blank. That gives me a raw score of the form $+n-m$ for each script.

When awarding \checkmark or \times I will carry forward some doubts from one observation to the next so that after I have given two \checkmark in a row based on marginal ~~clarity~~ clarity of explanation I get tough on the next.

At the end I have a table to map $+n-m$ scores onto "marks" in a way that leads to a plausible overall distribution...

The latex lost " $\phi \dots n$ " after printf but that did not hurt the question.

There are distinctly over 20 "glitches".

Note that one of the last lectures in the course had a class-wide "bug hunt"...

I had feared this might be an abomination to mark but in fact it admits a simpler marking scheme than pessimism would suggest & gives a broad spread of marks!