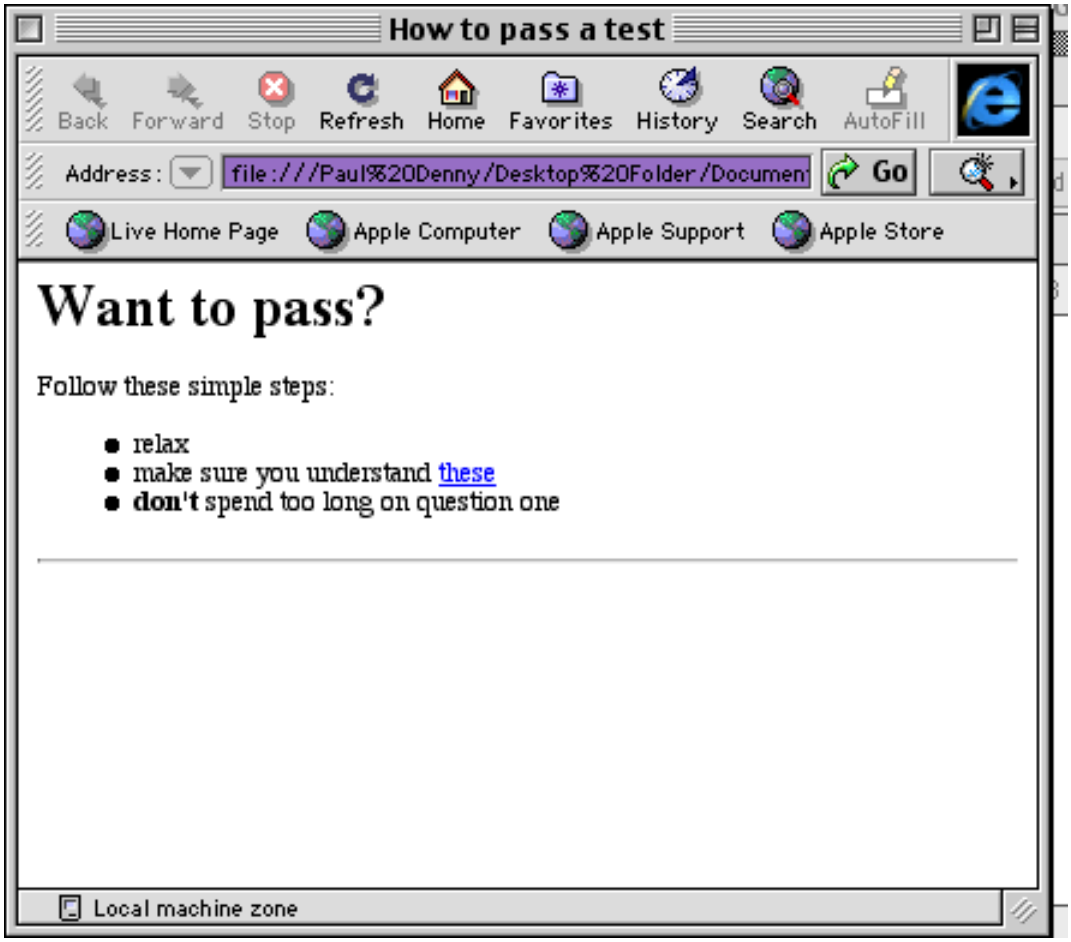
415.101FC PRINCIPLES OF PROGRAMMING

ANSWER BOOK

Test - Tuesday 11th April 6:30pm-8:00pm

1. Draw the output.



2. Corrected program (with changes CIRCLED):

1) Word static missing from main() declaration

2) variable toss is not declared

3) the = sign should be == in the if statement

4) semicolon missing from first println() statement

5) Logic error. Should be: toss = (int)(Math.random()*2);

3. Complete the paint() method

```
import java.awt.*;
import java.applet.*;
public class ScaleMe extends Applet {
public void paint(Graphics g) {
        final int SIZE = 50;
        g.drawRect(SIZE, SIZE, SIZE, SIZE);
        g.drawOval(0, 0, 3*SIZE, 3*SIZE);
        g.drawLine(SIZE, SIZE, 2*SIZE, 2*SIZE);
        g.drawLine(SIZE, 2*SIZE, 2*SIZE, SIZE);
}
```

4. Output from class Types:

a) 2.0

b) 4.5

c) 11.0

d) 2

e) 8

5. Output from class MethodTester:

From first:1

From first:2

From second:2

From first:3

From first:4

From second:4

From third:4

From init:4

6. Complete the FillBeaker applet.:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class FillBeaker extends Applet implements ActionListener {
        /** Left side of the beaker. */
        static final int LEFT = 40;
        /** Right side of the beaker . */
        static final int RIGHT = 100;
        /** Top of the beaker . */
        static final int TOP = 20;
        /** Bottom of the beaker . */
        static final int BOTTOM = 120;
        /** Left position of the buttons. */
        static final int BUTTON_LEFT = 140;
        /** Width of the buttons. */
        static final int BUTTON_WIDTH = 50;
        /** Height of the buttons. */
        static final int BUTTON_HEIGHT = 20;
        /** Top of the "more" button. */
        static final int MORE_TOP = 40;
        /** Top of the "less" button. */
        static final int LESS_TOP = 70;
        public void init() {
                setLayout(null);
                more = new Button("more");
                more.setBounds(BUTTON_LEFT, MORE_TOP, BUTTON_WIDTH,
                BUTTON_HEIGHT);
                add(more);
                more.addActionListener(this);
                less = new Button("less");
                less.setBounds(BUTTON_LEFT, LESS_TOP, BUTTON_WIDTH, BUTTON_HEIGHT);
```

```
            add(less);
            less.addActionListener(this);
            goo = 50;
        }
        public void actionPerformed(ActionEvent e) {
            if (e.getSource() == more && goo < 100)
                goo += 10;
            else if (e.getSource() == less && goo > 0)
                goo -= 10;
            repaint();
        }
        public void paint(Graphics g) {
            int topOfGoo;
            int width;
            g.setColor(Color.black);
            g.drawLine(LEFT, TOP, LEFT, BOTTOM);
            g.drawLine(LEFT, BOTTOM, RIGHT, BOTTOM);
            g.drawLine(RIGHT, TOP, RIGHT, BOTTOM);
            g.setColor(Color.pink);
            topOfGoo = BOTTOM - goo;
            width = RIGHT - LEFT - 1;
            g.fillRect(LEFT + 1, topOfGoo, width, goo);
        }
}
```

7. Complete the isInsideRect method:

```
        private boolean isInsideRect(int mouseX, int mouseY, int rectX, int rectY,
        int width, int height) {
            return rectX < mouseX &&
                mouseX < rectX + width &&
                rectY < mouseY &&
                mouseY < rectY + height;
```

```
        }
```

8. Complete the VerticalLine class:

```
import java.awt.*;

public class VerticalLine {

        private int x;

        private int length;

        public VerticalLine(int xPos, int lineLength) {

                x = xPos;

                length = lineLength;

        }

        public void draw(Graphics g) {

                final int BOTTOM = 100;

                g.setColor(Color.red);

                g.drawLine(x, BOTTOM, x, BOTTOM - length);

        }

}
```