Quiz: Java Bytecode

Java Bytecode is:

a) Machine code for the particular machine on which the Java source code was compiled.

b) The binary code Java uses to store bytes.

c) The hexadecimal code Java uses to store bytes.

d) Machine code for a Java Virtual Machine.

e) A device that bites, chews and spits Java code.
Quiz: Graphics Supplement 1.4

1) import javax.swing.JApplet;
2) import java.awt.Graphics;
3) public class Toy_1_4 extends JApplet
4) { public void paint(Graphics canvas)
5) { // drawOval(x, y, width, height);
6) canvas.drawOval(50, 50, 25, 25);
7) canvas.drawOval(100, 50, 25, 25);
8) canvas.drawOval(150, 50, 25, 25);
9) canvas.drawOval(200, 50, 25, 25);
10) }

(a) (b) (c) (d)

drawOval(x, y, width, height)

1) canvas.drawOval(50, 50, 25, 25);

drawOval is a method that requires four arguments.

Since width = height, the "oval" is a circle.

2) canvas.drawOval(100, 50, 25, 25);
3) canvas.drawOval(150, 50, 25, 25);
4) canvas.drawOval(200, 50, 25, 25);
Computer Coordinate System

A pixel is a square of light on a computer display.

Window with 200 x 200 pixel drawing area.

12 x 12 pixel segment of drawing area.
Background Color and Shapes

1) import javax.swing.JApplet;
2) import java.awt.Color;
3) import java.awt.Graphics;
4) public class Toy_1_4c extends JApplet
5) {
6)   public void paint(Graphics canvas)
7)   {
8)     canvas.setColor(Color.YELLOW);
9)     canvas.fillRect(0,0,200,200);
10)    canvas.setColor(Color.BLUE);
11)    //drawLine(int x1, int y1, int x2, int y2)
12)    canvas.drawLine(10, 50, 10, 180);
13)    canvas.drawLine(10, 50, 50, 180);
14)    canvas.drawLine(10, 50, 100, 180);
15)    canvas.setColor(Color.BLUE);
16)    canvas.drawLine(10, 50, 10, 180);
17)    canvas.drawLine(10, 50, 50, 180);
18)    canvas.drawLine(10, 50, 100, 180);
19)   }
20) }

Anti-Aliasing

Anti-aliasing is a transformation that tries to reduce artifacts caused by drawing on a grid.

Parallel lines drawn with Java. The second was drawn with anti-aliasing on.
Parallel Lines without/with Anti-Aliasing

```java
import javax.swing.JApplet;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
public class Toy_1_4b extends JApplet
{
    public void paint(Graphics canvas)
    {
        canvas.setColor(Color.BLUE);
        canvas.drawLine(0, 0, 166, 199);
        Graphics2D canvas2D = (Graphics2D) canvas;
        canvas2D.setRenderingHint(
            RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);
        canvas.drawLine(10, 0, 176, 199);
    }
}
```

Java: Applet verses Application

Java Application:
- Every application must have at least one class with
  `public static void main(String[] args)`
- Before an application can display graphics, it must create and display a window.

Java Applet:
- An applet runs inside a web browser window.
- Every applet starts with a graphics area given to it by the browser.
- Applets *can* create more windows, but usually they don't.
- Usually, applets are used for relatively simple programs.
Applets

A web page includes an applet by using the applet tag. For example:

```
<applet code="myApplet.class"
       width=300 height=200>
</applet>
```

The browser automatically calls:

- `init()`: called when the applet starts.
- `paint(Graphics canvas)`: called each time the browser thinks the applet needs to be repainted.
- Note: If you don't one of these, you will inherit an empty one from the `JApplet` class.

Overriding Applet's `init()` method

```
1) import javax.swing.JApplet;
2) import java.awt.Graphics;
3)
4) public class Toy_1_4_Applet2 extends JApplet
5) {
6)   public void init()
7)   { this.setSize(500,300); 
8)     }
9) }
10) public void paint(Graphics canvas)
11) { canvas.drawOval(100, 50, 25, 25);
12)   canvas.drawOval(100, 50, 50, 50);
13) }
14) }
```

Called once: When applet starts

Called every time repaint is needed
import javax.swing.JApplet;
import java.awt.Color;
import java.awt.Graphics;

public class Toy_1_4 extends JApplet {

    public void paint(Graphics canvas){
        Color periwinkle = new Color(204, 204, 255);
        Color pumpkin = new Color(255, 117, 24);

        canvas.setColor(periwinkle);
        canvas.fillOval(50, 50, 50, 50);

        canvas.setColor(pumpkin);
        canvas.fillOval(100, 50, 50, 100);
    }
}
How to Read Java Documentation

java.awt.Color Constructor:

Color(int red, int green, int blue)

Syntax Example:

canvas.drawOval( 50,50,25,25);
canvas.drawOval(100,50,50,100);

Applet in Action

1) import javax.swing.JApplet;
2) import java.awt.Graphics;
3) public class Toy_1_4 extends JApplet
4) {
5)    public void paint(Graphics canvas)
6)    {
7)      System.out.println("called paint()");
8)      canvas.drawOval(50,50,25,25);
9)      canvas.drawOval(100,50,50,100);
10)    }
11}
drawOval(int left, int top, int width, int height)
fillOval(int left, int top, int width, int height)
drawArc(int left, int top, int width, int height, int startAngle, int arcAngle)
drawRect(int left, int top, int width, int height)
fillRect(int left, int top, int width, int height)
drawLine(int x1, int y1, int x2, int y2)
setColor(java.awt.Color myColor)