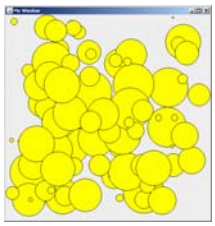
Lecture 22

Drawing objects

Section 16.4
Drawing Objects

Draw random circles

Consider the following picture of a collection of circles:



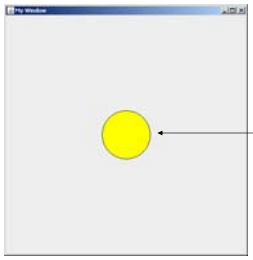
Each circle has a random radius between **2** and **50** (inclusive).

Each circle is located in a random position but entirely within the bounds of the window which is **500** pixels high and **500** pixels wide.

How could we write this program?

Draw a circle

Lets start by drawing one circle, like this:

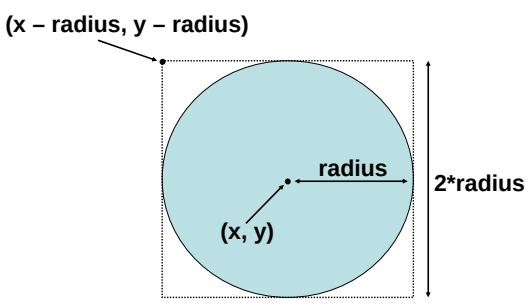


this circle is centered at position (250, 250), and has a radius of 50

```
public class MyJPanel extends JPanel {

        < define instance variables >

        public MyJPanel() {

                < initialise instance variables >

        }

        public void paintComponent(Graphics g){
                super.paintComponent(g);

                < display drawing based on instance variables >

        }
}
```

```
public class MyJPanel extends JPanel {

        < define instance variables >

        public MyJPanel() {

                radius = 50;
                x = 250;
                y = 250;
        }

        public void paintComponent(Graphics g){
                super.paintComponent(g);

                < display drawing based on instance variables >

        }
}
```

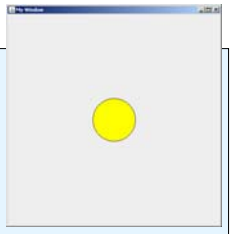
```
public class MyJPanel extends JPanel {

        private int radius;
        private int x;
        private int y;

        public MyJPanel() {

                radius = 50;
                x = 250;
                y = 250;
        }

        public void paintComponent(Graphics g){
                super.paintComponent(g);

                < display drawing based on instance variables >

        }
}
```




```
g.fillOval(x – radius, y – radius, 2*radius, 2*radius);
```

```
public class MyJPanel extends JPanel {

        private int radius;
        private int x;
        private int y;

        public MyJPanel() {

                radius = 50;
                x = 250;
                y = 250;
        }

        public void paintComponent(Graphics g){
                super.paintComponent(g);


                < display drawing based on instance variables >

        }
}
```

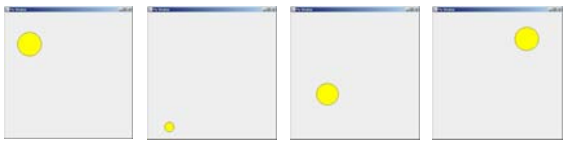
```
public class MyJPanel extends JPanel {

        private int radius;
        private int x;
        private int y;

        public MyJPanel() {

                radius = 50;
                x = 250;
                y = 250;
        }

        public void paintComponent(Graphics g){
                super.paintComponent(g);

                g.setColor(Color.yellow);
                g.fillOval(x - radius, y - radius, 2*radius, 2*radius);
                g.setColor(Color.black);
                g.drawOval(x - radius, y - radius, 2*radius, 2*radius);
        }
}
```



```
public class MyJPanel extends JPanel {

        private int radius;
        private int x;
        private int y;

        public MyJPanel() {

                radius = 50;
                x = 250;
                y = 250;
        }

        public void paintComponent(Graphics g){
                super.paintComponent(g);

                g.setColor(Color.yellow);
                g.fillOval(x - radius, y - radius, 2*radius, 2*radius);
                g.setColor(Color.black);
                g.drawOval(x - radius, y - radius, 2*radius, 2*radius);
        }
}
```

Random position and size

How do we generate a random radius value, and a random position for the circle?



Radius: between 2 and 50 inclusive
Position: inside the window (500 x 500 pixels)

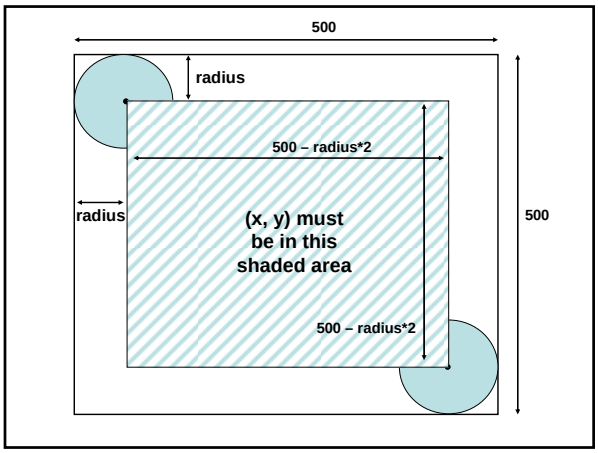
Random radius

How do we generate a random integer value between 2 and 50 inclusive?

```
radius = (int)(Math.random()*49) + 2;
```

How do we generate random values for x and y?

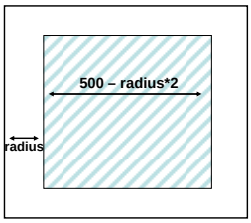
```
x = ????;
y = ????;
```



Random position

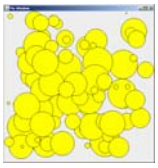
How do we generate random values for x and y?

```
x = (int)(Math.random()*(500-radius*2) + radius);
y = (int)(Math.random()*(500-radius*2) + radius);
```



```
public class MyJPanel extends JPanel {

        private int radius;
        private int x;
        private int y;

        public MyJPanel() {

                radius = (int)(Math.random()*49) + 2;
                x = (int)(Math.random()*(500-radius*2) + radius);
                y = (int)(Math.random()*(500-radius*2) + radius);
        }

        public void paintComponent(Graphics g){
                super.paintComponent(g);

                g.setColor(Color.yellow);
                g.fillOval(x - radius, y - radius, 2*radius, 2*radius);
                g.setColor(Color.black);
                g.drawOval(x - radius, y - radius, 2*radius, 2*radius);
        }
}
```

A Circle class



Now, how do we display lots of random circles?

We need to store the radius and position of each circle.

We can either use three separate arrays:

```
public class MyJPanel extends JPanel {

        private int[] radiusValues;
        private int[] xValues;
        private int[] yValues;
```

Or a single array of Circle objects:

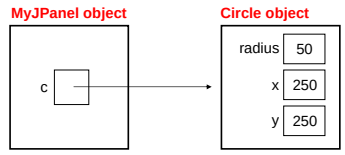
```
public class MyJPanel extends JPanel {

        private Circle[] circles;
```

```
import java.awt.*;

public class Circle {

    < define instance variables >


    public Circle(int radius, int x, int y) {

            < initialise instance variables >

    }

    public void draw(Graphics g) {


            < display drawing based on instance variables >

    }
}
```

```
import java.awt.*;

public class Circle {

    private int radius;
    private int x;
    private int y;

    public Circle(int radius, int x, int y) {

            < initialise instance variables >

    }

    public void draw(Graphics g) {


            < display drawing based on instance variables >

    }
}
```

```
import java.awt.*;

public class Circle {

    private int radius;
    private int x;
    private int y;

    public Circle(int radius, int x, int y) {
        this.radius = radius;
        this.x = x;
        this.y = y;
    }

    public void draw(Graphics g) {

        < display drawing based on instance variables >

    }
}
```

```
import java.awt.*;

public class Circle {

    private int radius;
    private int x;
    private int y;

    public Circle(int radius, int x, int y) {
        this.radius = radius;
        this.x = x;
        this.y = y;
    }

    public void draw(Graphics g) {
        g.setColor(Color.yellow);
        g.fillOval(x - radius, y - radius, 2*radius, 2*radius);
        g.setColor(Color.black);
        g.drawOval(x - radius, y - radius, 2*radius, 2*radius);
    }
}
```

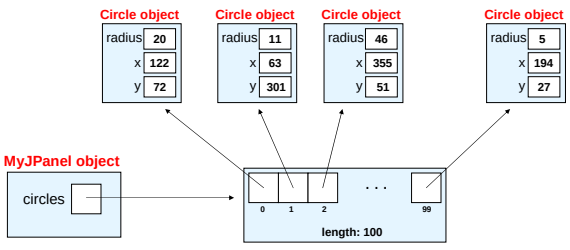



```
public class MyJPanel extends JPanel {

    private Circle c;

    public MyJPanel() {
        c = new Circle(50, 250, 250);
    }

    public void paintComponent(Graphics g){
        super.paintComponent(g);

        c.draw(g);
    }
}
```

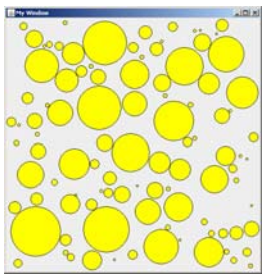
```
public class MyJPanel extends JPanel {

    private Circle[] circles;

    public MyJPanel() {

        circles = new Circle[100];

        for (int i = 0; i < circles.length; i++) {
              int x, y, radius;
              radius = (int)(Math.random()*49) + 2;
              x = (int)(Math.random()*(500-radius*2) + radius);
              y = (int)(Math.random()*(500-radius*2) + radius);
              circles[i] = new Circle(radius, x, y);
        }
    }

    public void paintComponent(Graphics g){
        super.paintComponent(g);

        for (int i = 0; i < circles.length; i++) {
              circles[i].draw(g);
        }
    }
}
```

Lots of circles

We can visualise the array of Circle objects as follows:



Challenge?



exactly the same as before, but ensure that none of the circles overlap