

## Week 4, Lecture 2

### Working with Arrays



1

## Announcements

- **1st Assignment** is due **today**: deadline is **4pm!**
- the deadline for the **first three lab assessments** is **next week (Week 5)**.

2

## This Week

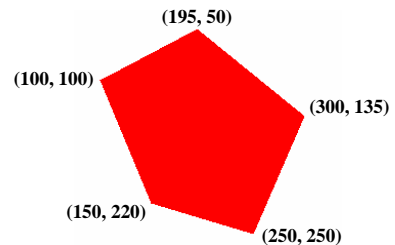
### *Java Genesis:*

–**Ch5: Arrays** (Section 5.4 is optional)

**Assignment 1** due 4pm today!

3

## A Polygon



4

## The Shape class

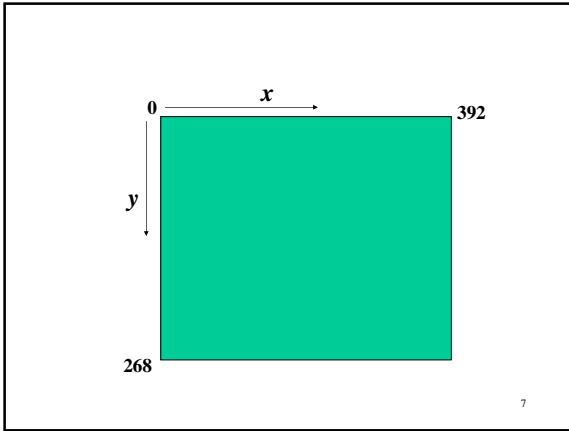
When sent the message `create()` an empty window opens.

When sent the message `draw(xs, ys)` where `xs` and `ys` are integer arrays of the same length, a solid red polygon is drawn with vertices `(xs[0], ys[0])`, `(xs[1], ys[1])`, `(xs[2], ys[2])`, ....

5

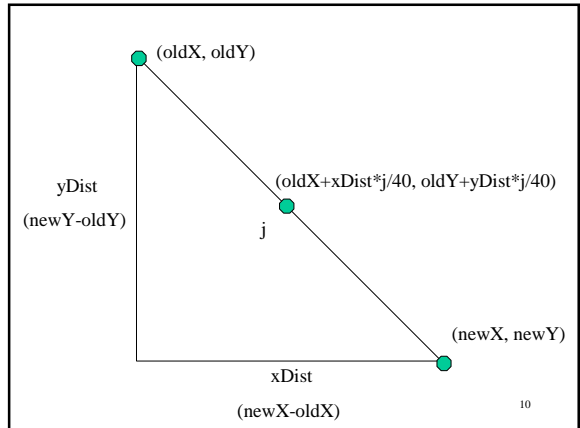
```
public class DrawShape {  
    public static void main (String [ ] args) {  
        Shape.create();  
        int [ ] xCoords = {195, 300, 250, 150, 100};  
        int [ ] yCoords = {50, 135, 250, 220, 100};  
        Shape.draw(xCoords, yCoords);  
    }  
}
```

6



```
public class DrawShape {
    public static void main (String [ ] args) {
        Shape.create();
        int num = 40;
        int [ ] xCoords = new int [num];
        int [ ] yCoords = new int [num];
        for (int i=0; i<num; i++) {
            xCoords[i] = (int)(392*Math.random());
            yCoords[i] = (int)(268*Math.random());
        }
        Shape.draw(xCoords, yCoords);
    }
}
```

```
import genesis.*;
public class LostCircle {
    public static void main (String [ ] args) {
        CircleFigure.create();
        int newX, newY, oldX, oldY, xDist, yDist, currentX, currentY;
        for (int i=1; i<=50; i++) {
            newX = (int)(346*Math.random())+25;
            newY = (int)(216*Math.random())+25;
            oldX = CircleFigure.getXCentre();
            oldY = CircleFigure.getYCentre();
            xDist = newX - oldX;
            yDist = newY - oldY;
            for (int j=1; j<=40; j++) {
                Delay.milliSeconds(25);
                currentX = (int)(oldX+xDist*j/40.0);
                currentY = (int)(oldY+yDist*j/40.0);
                CircleFigure.moveTo(currentX, currentY);
            }
        }
    }
}
```



```
import genesis.*;
public class LostShape {
    public static void main (String [ ] args) {
        int [ ] xCoords = {195, 300, 250, 150, 100};
        int [ ] yCoords = {50, 135, 250, 220, 100};
        Shape.create();
        Shape.draw(xCoords, yCoords);
        int newX, newY, oldX, oldY, xDist, yDist;
        while (true) {
            newX = (int)(392*Math.random());
            newY = (int)(268*Math.random());
            oldX = xCoords[2];
            oldY = yCoords[2];
            xDist = newX - oldX;
            yDist = newY - oldY;
            for (int j=1; j<=40; j++) {
                Delay.milliSeconds(25);
                xCoords[2] = (int)(oldX+xDist*j/40.0);
                yCoords[2] = (int)(oldY+yDist*j/40.0);
                Shape.draw(xCoords, yCoords);
            }
        }
    }
}
```

```
import genesis.*;
public class LostShape {
    public static void main (String [ ] args) {
        int num = 40;
        int [ ] xCoords = new int [num];
        int [ ] yCoords = new int [num];
        for (int i=0; i<num; i++) {
            xCoords[i] = 200;
            yCoords[i] = 150;
        }
        Shape.create();
        Shape.draw(xCoords, yCoords);
        int [ ] newX = new int [num];
        int [ ] newY = new int [num];
        int [ ] oldX = new int [num];
        int [ ] oldY = new int [num];
        int [ ] xDist = new int [num];
        int [ ] yDist = new int [num];
    }
}
```

```

while (true) {
    for (int i=0; i<num; i++) {
        newX[i] = (int)(392*Math.random());
        newY[i] = (int)(268*Math.random());
        oldX[i] = xCoords[i];
        oldY[i] = yCoords[i];
        xDist[i] = newX[i] - oldX[i];
        yDist[i] = newY[i] - oldY[i];
    }
    for (int j=1; j<=40; j++) {
        Delay.miliseconds(50);
        for (int i=0; i<num; i++) {
            xCoords[i] = (int)(oldX[i]+xDist[i]*j/40.0);
            yCoords[i] = (int)(oldY[i]+yDist[i]*j/40.0);
        }
        Shape.draw(xCoords, yCoords);
    }
}
}
}

```

13

```

import genesis.*;
import java.awt.*;
public class RandomColours {

    public static void main (String [ ] args) {
        CircleFigure.create( );
        CircleFigure.setRadius(100);
        int rand;
        while (true) {
            Delay.miliseconds(500);
            rand = (int)(5*Math.random( ));
            switch (rand) {
                case 0: CircleFigure.setColor(Color.red); break;
                case 1: CircleFigure.setColor(Color.blue); break;
                case 2: CircleFigure.setColor(Color.green); break;
                case 3: CircleFigure.setColor(Color.magenta); break;
                case 4: CircleFigure.setColor(Color.yellow); break;
            }
        }
    }
}

```

14

```

import genesis.*;
import java.awt.*;
public class RandomColours {

    public static void main (String [ ] args) {
        Color [ ] colours =
            {Color.red, Color.blue, Color.green,
            Color.magenta, Color.yellow};
        CircleFigure.create( );
        CircleFigure.setRadius(100);
        int rand;
        while (true) {
            Delay.miliseconds(500);
            rand = (int)(5*Math.random( ));
            CircleFigure.setColor(colours[rand]);
        }
    }
}

```

15