G22.2210

Programming Languages: PL

B. Mishra
New York University.

Lecture # 12

—Slide 1—

Java Examples

Objects and Classes

```
/* FILE: Rectangle.java */
class Rectangle {
  private int x;
  private int y;
  private int width;
  private int height;

  public Rectangle(int x, int y, int width, int height){
    this.x = x;
    this.y = y;
    this.width = width;
    this.height = height;
  }
```


—Slide 2—

Objects and Classes, Contd

```
  public void draw(){
    System.out.println("Rectangle:"+x +","+y +
                       ","+width +","+height);
  }

  public void draw(Graphics g){
    g.drawRect(x, y, width, height);
  }

  public void advance(int x, int y){
    this.x = x;
    this.y = y;
  }

  public static void main(String args[]){
    Rectangle r = new Rectangle(5, 5, 100, 200);
    Graphics g = new Graphics();

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


—Slide 3—

Java Examples, Contd

Inheritance

```
import java.awt.*;

class Shape {
  protected Color color;
  protected int x;
  protected int y;
  protected Shape(Color c, int x, int y){
    color = c;
    this.x = x;
    this.y = y;
  }

  public abstract void draw();
  public abstract void draw(Graphics g);

  public void advance(int x, int y){
    this.x = x;
    this.y = y;
  }
}
```

—Slide 4—

Inheritance, Rectangle

```
class Rectangle extends Shape {
  private int width;
  private int height;
  public Rectangle(Color c, int x, int y,
                   int height, int width) {
    super(c, x, y);
    this.width = width;
    this.height = height;
  }

  public void draw(){
    System.out.println("Rectangle:"+x +","+y +
                       ","+width +","+height);
  }

  public void draw(Graphics g){
    g.drawRect(x, y, width, height);
  }
}
```


—Slide 5—

Inheritance, Circle

```
class Circle extends Shape {
  private int radius;
  public Circle(Color c, int x, int y,
                int radius) {
    super(c, x, y);
    this.radius = radius;
  }

  public void draw(){
    System.out.println("Circle:"+x +","+y +
                       ","+radius);
  }

  public void draw(Graphics g){
    g.drawCirc(x, y, width, height);
  }
}
```


—Slide 6—

Inheritance, GraphicsProgram

```
public GraphicsProgram{
  public static void main(String args[]){
    Shape r = new Rectangle(Color.red, 5, 5, 100, 200);
    Shape c = new Circle(Color.green, 20, 30, 100);
    Graphics g = new Graphics();

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


—Slide 7—

Java Examples, Contd

Interface and Threads

```
package java.lang

public interface Runnable {
  void run();
}
```

- Any class that implements `Runnable` should define a `run` method.
- The constructor for the `Thread` class takes an instance of the `Runnable` interface and invokes the `run` method when the thread is started.
- When the `run` method returns, the thread will automatically exit.

—Slide 8—

Java Examples, Contd

Animation

```
class Animation {
  protected Applet app;

  protected void init(Applet app){
    this.app = app;
  }

  public abstract void advance();

  public abstract void paintFrame(Graphics g);
}
```


—Slide 9—

Java Examples, Contd

Rectangle Animation

```
class RectangleAnimation extends Animation {
  private Shape r = new Rectangle(Color.blue, 0, 0, 50, 50);

  public void advance() {
    Rectangle bounds = app.bounds;
    r.advance((Math.random() * 1000) % bounds.width,
              (Math.random() * 1000) % bounds.height);
  }

  public void paintFrame(Graphics g) {
    r.draw(g);
  }
}
```


—Slide 10—

Java Examples, Contd

Animation—Bouncing Rectangles (`init` and `start`)

```
public class AnimationApplet extends Applet
                      implements Runnable {
  Thread animator;
  Animation animation;

  public void init() {
    animator = new Thread(this);
    animation = new RectangleAnimation();
    animation.init(this)
  }

  public void start() {
    if (animator.isAlive()) {
      animator.resume();
    } else {
      animator.start();
    }
  }
```


—Slide 11—

Java Examples, Contd

Bouncing Rectangles (`stop`, `destroy`, `run`, `paint`)

```
  public void stop() {
    animator.suspend();
  }

  public void destroy() {
    animator.stop();
  }

  public void run() {
    while(true){
      repaint();
      Thread.sleep(500);
      animation.advance();
    }
  }

  public void paint(Graphics g) {
    animation.paintFrame(g);
  }
}
```

—Last Slide—

Features of the Java Example

- *Java Class and Objects*
- *Inheritance in Java*
- *Java Threads*
- *Java Interfaces*
- *Java Applet*

[End of Lecture #12]