Computer Programming Fundamentals

CS 152
Professor: Leah Buechley
TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza
Time: MWF 10:00-10:50am
https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/

ASSIGNMENT 3 DUE TODAY

QUIZ 1 GRADED

LEARN ISSUE

TODAY: CLASSES AND OBJECTS cont.

BALL CLASS

```
import java.awt.*;

public class Ball {
    Color color;
    int size;
    int xPosition;
    int yPosition;
    int xSpeed;
    int ySpeed;

    Ball () {
        color = Color.PINK;
        size = 50;
        xPosition = 100;
        yPosition = 100;
        xSpeed = 1;
        ySpeed = 1;
    }

    Ball (Color color, int size, int xPosition, int yPosition) {
        this.color = color;
        this.size = size;
        this.xPosition = xPosition;
        this.yPosition = yPosition;
        xSpeed = 0;
        ySpeed = 0;
    }

    public static void main(String[] args) {
        Ball ball;
        ball = new Ball();
        ball.move();

        Ball ball2;
        ball2 = new Ball(Color.BLUE, 100, 10, 500);
        System.out.println("ball2 xPosition: " +ball2.xPosition);
        ball2.setSpeed(100,100);
        ball2.move();
        System.out.println("ball2 xPosition after move: " +ball2.xPosition);
    }

    void move() {
        xPosition = xPosition + xSpeed;
        yPosition = yPosition + ySpeed;
    }

    void setSpeed (int xSpeed, int ySpeed) {
        this.xSpeed = xSpeed;
        this.ySpeed = ySpeed;
    }
}
```

STRUCTURE

```
import java.awt.*;

public class Ball {
    Color color;
    int size;
    int xPosition;
    int yPosition;
    int xSpeed;
    int ySpeed;

    Ball () {
        color = Color.PINK;
        size = 50;
        xPosition = 100;
        yPosition = 100;
        xSpeed = 1;
        ySpeed = 1;
    }

    Ball (Color color, int size, int xPosition, int yPosition) {
        this.color = color;
        this.size = size;
        this.xPosition = xPosition;
        this.yPosition = yPosition;
        xSpeed = 0;
        ySpeed = 0;
    }

    public static void main(String[] args) {
        Ball ball;
        ball = new Ball();
        ball.move();

        Ball ball2;
        ball2 = new Ball(Color.BLUE, 100, 10, 500);
        System.out.println("ball2 xPosition: " +ball2.xPosition);
        ball2.setSpeed(100,100);
        ball2.move();
        System.out.println("ball2 xPosition after move: " +ball2.xPosition);
    }

    void move() {
        xPosition = xPosition + xSpeed;
        yPosition = yPosition + ySpeed;
    }

    void setSpeed (int xSpeed, int ySpeed) {
        this.xSpeed = xSpeed;
        this.ySpeed = ySpeed;
    }
}
```

variable declarations
“instance” variables

constructors
have same name as class
have no return type

each constructor has
different input parameters

main method

code that runs when you
run the program

other class methods

TODAY: RETURNING TO GRAPHICS SO WE CAN PLAY WITH OUR BALL

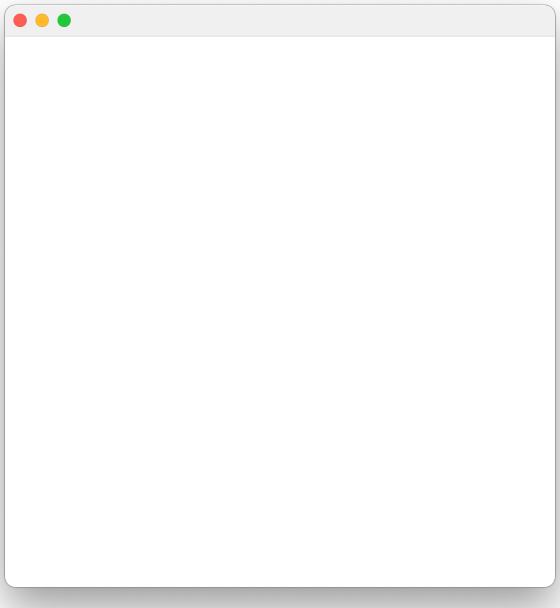
ONE WAY TO WORK WITH GRAPHICS
JFrame, JPanel, and Graphics

- These are all classes that are built into Java
- Someone else wrote them, but we can use them
- They are in the javax.swing and java.awt “packages”
- A package is a collection of classes

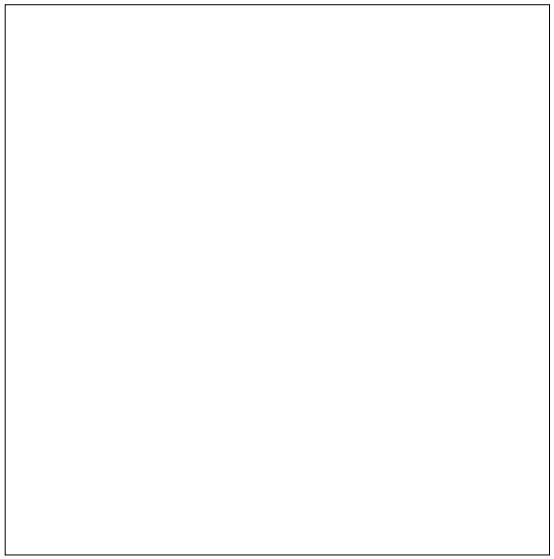
ONE WAY TO WORK WITH GRAPHICS
JFrame, JPanel, and Graphics

- JFrame = the window
- JPanel = the surface where we draw things
- Graphics = the collection of things that are drawn

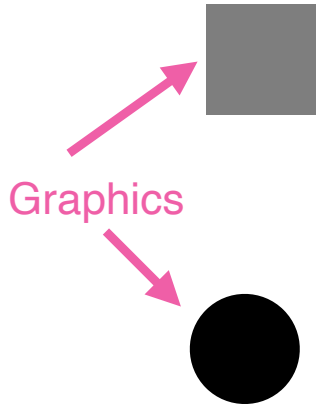
JFrame, JPanel, and Graphics



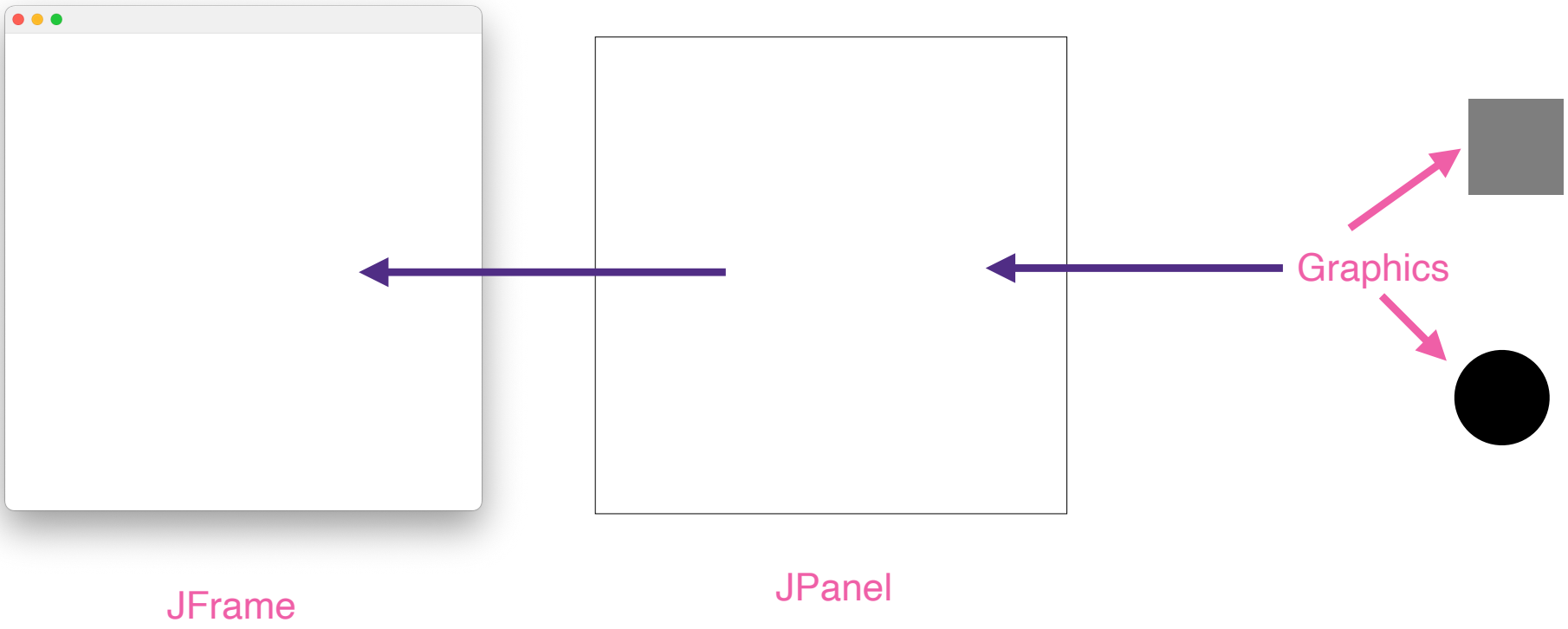
JFrame



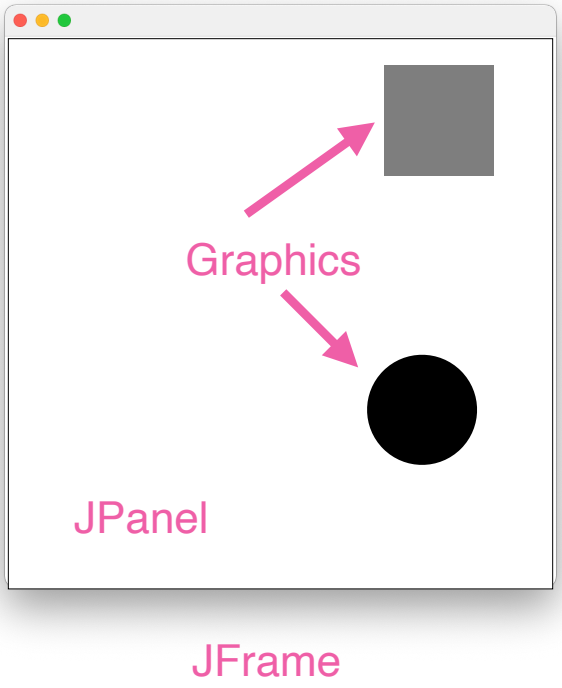
JPanel



PUTTING IT TOGETHER



PUTTING IT TOGETHER



CREATE A FRAME
CREATE A PANEL
DRAW GRAPHICS ON THE PANEL

OPEN IntelliJ
AND PROJECT FROM LAST CLASS

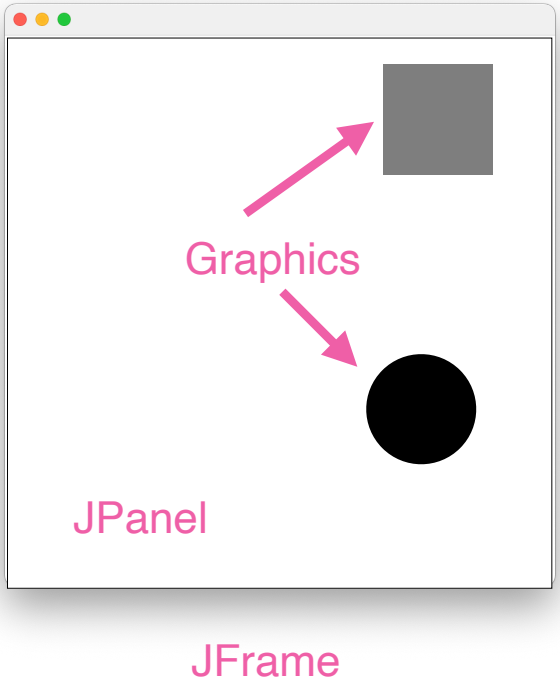
CREATE A NEW CLASS CALLED MyFrame.java

IMPORT THE PACKAGES WE NEED: javax.swing and java.awt

```
import javax.swing.*;
import java.awt.*;

public class myFrame {

}
```

the * means that all classes from those packages should be imported

THE FRAME



we want our MyFrame class to have a frame and a panel.

(Graphics will be part of the panel object.)

FIRST PART OF CLASS: VARIABLES

```
import javax.swing.*;
import java.awt.*;

public class myFrame {
    JFrame frame;
    JPanel panel;
}
```

we want our MyFrame class to have a frame and a panel.

NEXT ELEMENT OF A CLASS?

CONSTRUCTOR

```
import javax.swing.*;
import java.awt.*;

public class myFrame {
    JFrame frame;
    JPanel panel;

    myFrame() {
    }
}
```

NEXT ELEMENT OF A CLASS?

we don’t need any methods yet

FILL IN CONSTRUCTOR. HOW?

ASSIGN VALUES TO ALL VARIABLES

```
import javax.swing.*;
import java.awt.*;

public class myFrame {
    JFrame frame;
    JPanel panel;

    myFrame(JPanel panel) {
        frame = new JFrame();
        this.panel = panel;
    }
}
```

we want our MyFrame class to have a frame and a panel.

note: constructor takes a panel as an input.

NEED A FEW MORE THINGS

```
import javax.swing.*;
import java.awt.*;

public class myFrame {
    JFrame frame;
    JPanel panel;

    myFrame(JPanel panel) {
        frame = new JFrame();
        this.panel = panel;
        frame.add(panel);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}
```

add the panel to the frame
configure the frame

questions?

~~CREATE A FRAME~~

CREATE A PANEL

DRAW GRAPHICS ON THE PANEL

CREATE A NEW CLASS CALLED MyPanel.java

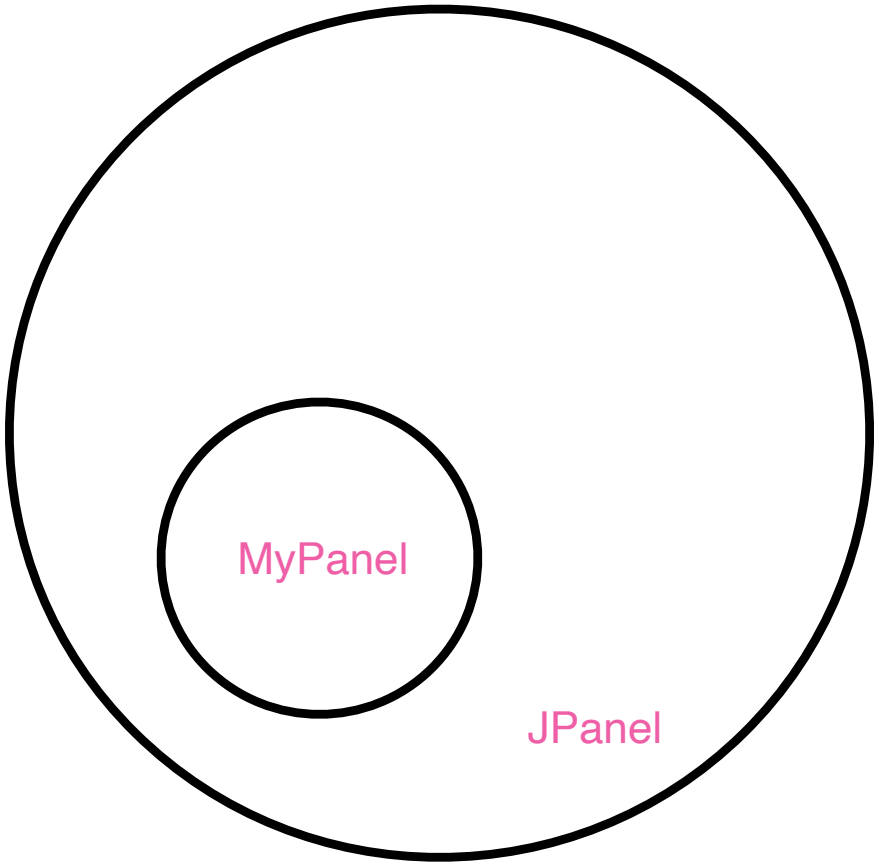
IMPORT THE PACKAGES WE NEED: javax.swing and java.awt

```
import javax.swing.*;
import java.awt.*;

public class myPanel {

}
```

JAVA INHERITANCE

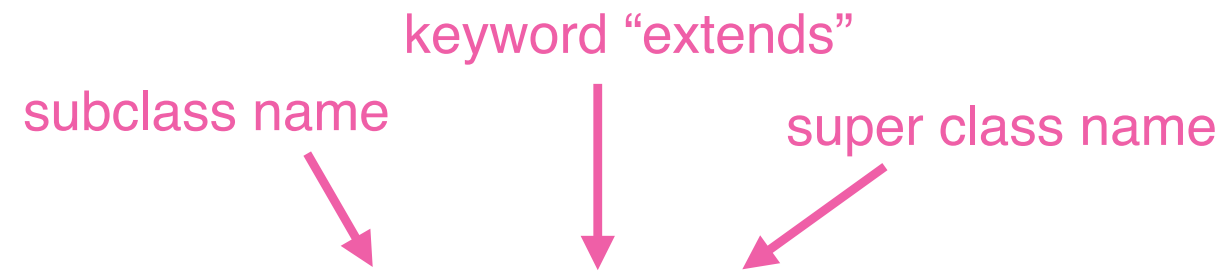
- We want our MyPanel class to create objects that are of type JPanel, with some special additional features
- We can make MyPanel a “subclass” of JPanel
- A way that object oriented programming supports code efficiency

JAVA INHERITANCE



- MyPanel is a “subclass” of JPanel
- JPanel is a “super” class of MyPanel
- All MyPanel objects are JPanel objects
- MyPanel objects contain and have access to all methods and variables defined in JPanel
- MyPanel class can define additional variables and methods that are not part of JPanel

CREATING A SUBCLASS




```
public class MyPanel extends JPanel {

}
```

questions?

FIRST PART OF CLASS: VARIABLES

```
import javax.swing.*;
import java.awt.*;

public class MyPanel extends JPanel {
    int width;
    int height;
}
```

width and height
will keep track of the
size of our panel

NEXT ELEMENT OF A CLASS?

CONSTRUCTOR

```
import javax.swing.*;
import java.awt.*;

public class MyPanel extends JPanel {
    int width;
    int height;

    MyPanel() {

    }
}
```

NEXT ELEMENT OF A CLASS?

METHODS

METHODS: A WAY TO DRAW STUFF
THE paintComponent METHOD

```
import javax.swing.*;
import java.awt.*;

public class MyPanel extends JPanel {
    int width;
    int height;

    MyPanel() {

    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
    }
}
```

we are redefining or "overriding" a method defined in JPanel

NOW GO BACK AND FILL STUFF IN

FILL IN CONSTRUCTOR

```
MyPanel() {
    width = 500;
    height = 500;
}
```

NEED A FEW MORE THINGS

```
MyPanel() {
    width = 500;
    height = 500;
    Dimension d = new Dimension(width, height);
    setPreferredSize(d);
    setVisible(true);
}
```

these are methods from JPanel
set the size
make the panel visible

questions?

~~CREATE A FRAME~~

~~CREATE A PANEL~~

DRAW GRAPHICS ON THE PANEL

WHAT DO WE NEED TO ADD TO USE OUR NEW CLASSES?

ADD A MAIN METHOD

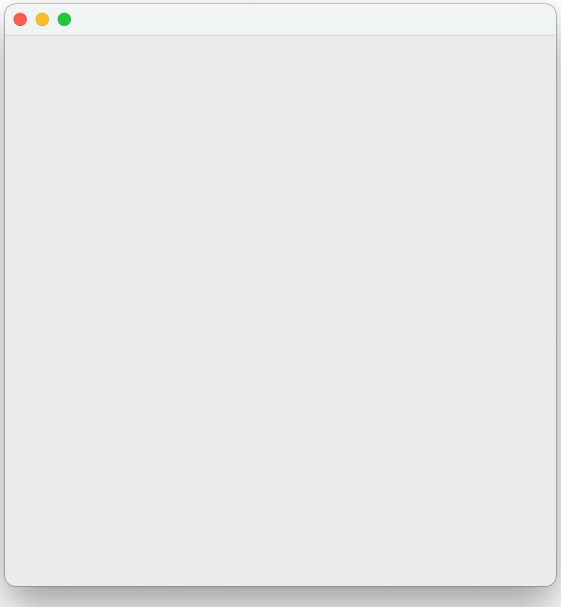
```
import javax.swing.*;
import java.awt.*;

public class MyPanel extends JPanel {
    int width;
    int height;

    MyPanel() {
        width = 500;
        height = 500;
        Dimension d = new Dimension(width, height);
        setPreferredSize(d);
        setVisible(true);
    }

    public static void main(String[] args) {

    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
    }
}
```

main method contains the code that actually runs the entry point

IN MAIN: CREATE A PANEL & FRAME

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
}
```

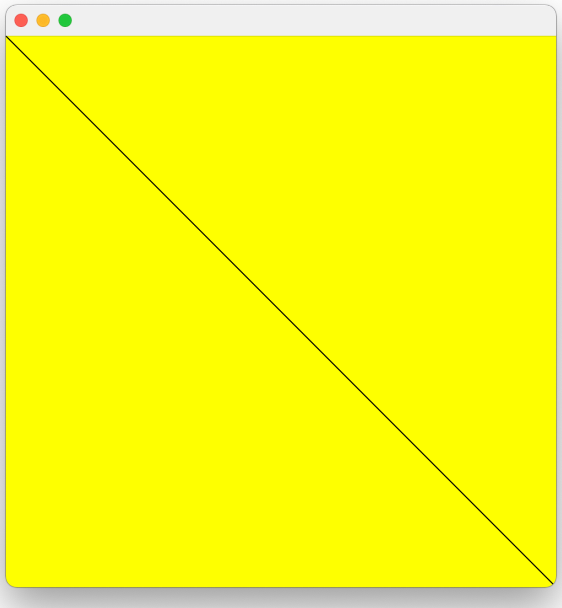
run the code to create an (empty) window



ADD CODE TO PaintComponent TO DRAW

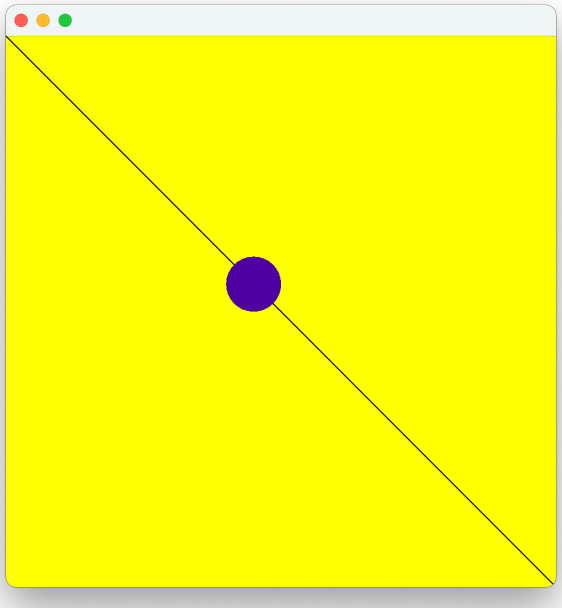
```
protected void paintComponent(Graphics g){
    super.paintComponent(g);
    setBackground(Color.YELLOW);
    g.drawLine(0,0,width,height);
}
```

add code *after* the super.paintComponent(g) line



CAN USE ALL OF THE DRAWING FEATURES YOU'RE FAMILIAR WITH

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(Color.YELLOW);
    g.drawLine(0,0,width,height);
    Color c = new Color(73, 11, 155, 255);
    g.setColor(c);
    g.fillOval(200,200,50,50);
}
```



~~CREATE A FRAME~~

~~CREATE A PANEL~~

~~DRAW GRAPHICS ON THE PANEL~~

questions?

NOW WE CAN ADD A BALL

ADD A BALL VARIABLE TO MyPanel

```
public class MyPanel extends JPanel {
    int width;
    int height;
    Ball ball;

    MyPanel() {
        width = 500;
        height = 500;
        Dimension d = new Dimension(width, height);
        setPreferredSize(d);
        setVisible(true);
    }

...
```

CREATE/INSTANTIATE THE BALL IN THE CONSTRUCTOR

```
public class MyPanel extends JPanel {
    int width;
    int height;
    Ball ball;

    MyPanel() {
        width = 500;
        height = 500;
        ball = new Ball(Color.BLACK, 100, 100,200);
        Dimension d = new Dimension(width, height);
        setPreferredSize(d);
        setVisible(true);
    }

...
```

HOW TO DRAW THE BALL?
WHERE TO DRAW THE BALL?

GOOD CODING PRACTICE: ANYTHING THAT A BALL DOES OR THAT IS DONE TO A BALL SHOULD HAPPEN IN THE BALL CLASS

GOOD CODING PRACTICE: ANYTHING THAT AN OBJECT DOES OR THAT IS DONE TO AN OBJECT SHOULD HAPPEN IN THAT CLASS

ADD A DRAW METHOD TO THE BALL CLASS

Thank you!

CS 152
Professor: Leah Buechley
TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza
Time: MWF 10:00-10:50am
https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/