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 1

- Graded
- Unless you had Learn problems or had to submit via email. We’ll have those up soon.
- Let us know if you have any questions

ASSIGNMENT 2

- Nice job!
- We will have grades back to you next week
- We will go over next week

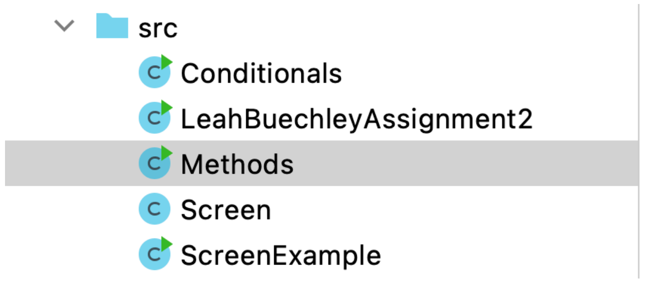
UNM DROP DEADLINE

- Friday 9/10
- Last chance to drop without paying tuition
- Will receive W on transcript
- If you haven’t done any assignments so far, you will be dropped
- I have emailed you if you are in danger. Please be in touch if you want to stay in the class!

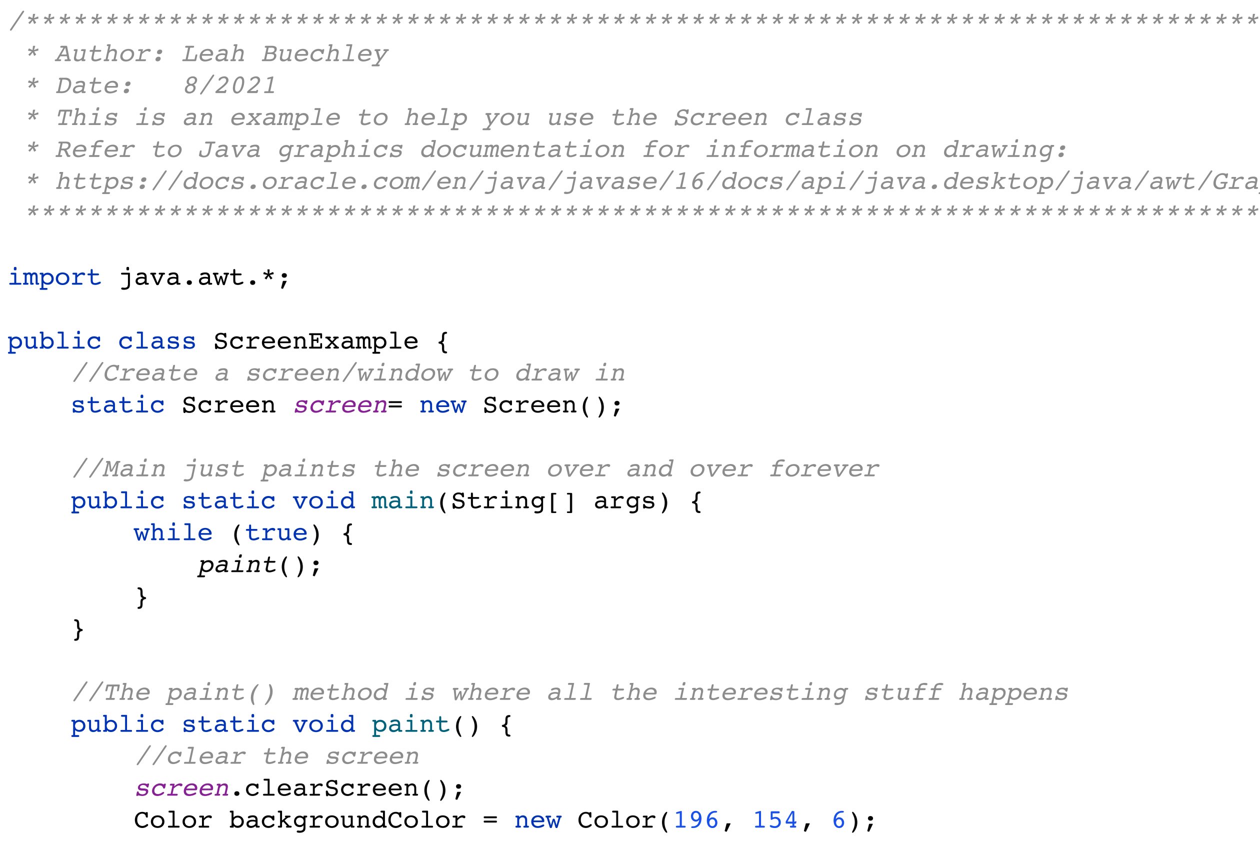
OPEN INTELLIJ & LAST WEEK’S PROJECT

CREATE A NEW JAVA CLASS FILE NAMED Methods.java

COPY & PASTE ScreenExample.java CODE FROM CLASS WEBSITE

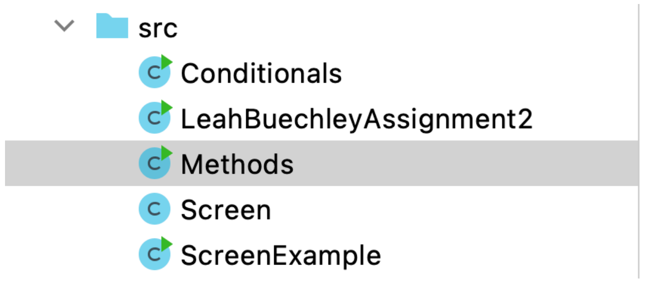


```
src
    Conditionals
    LeahBuechleyAssignment2
    Methods
    Screen
    ScreenExample
```

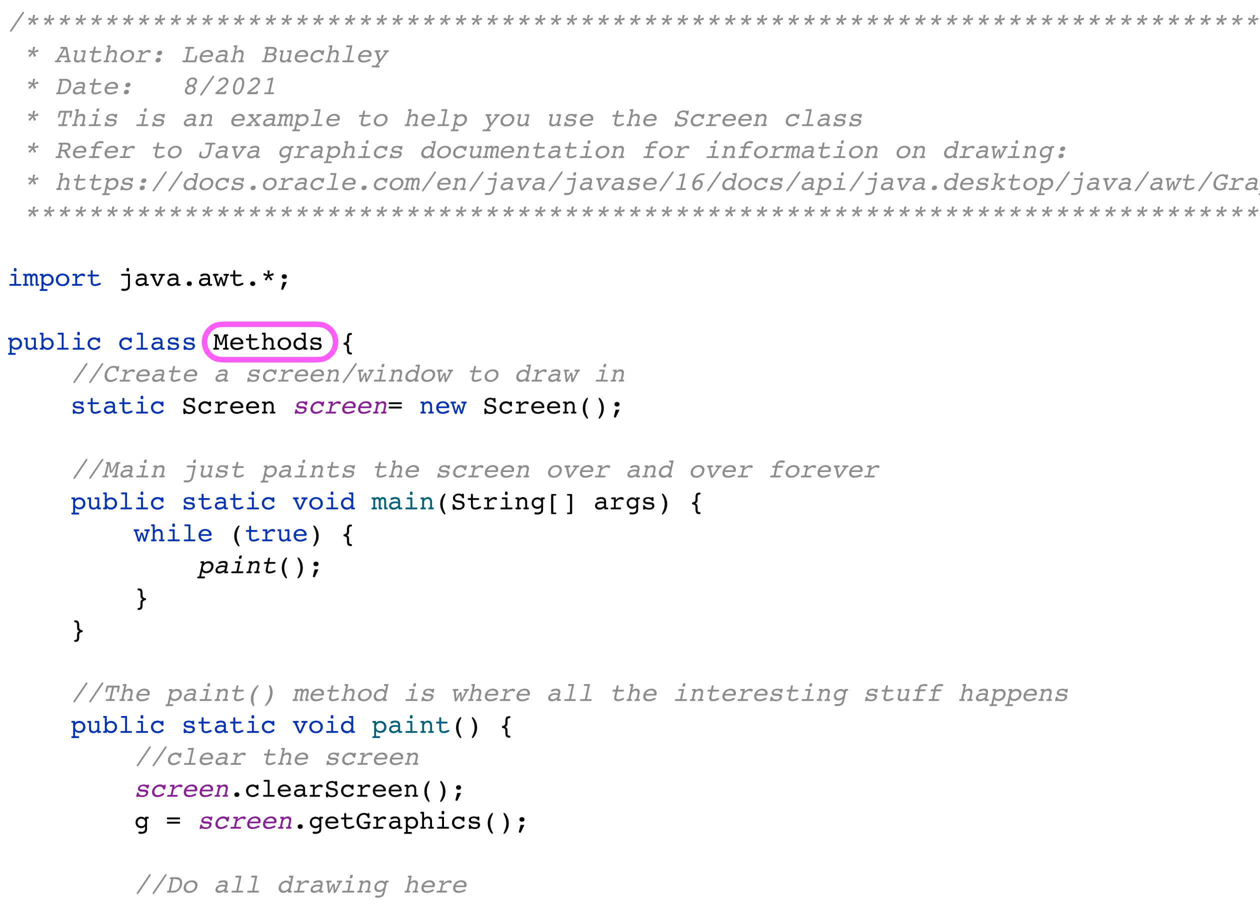


```
/***********************************************************************
 * Author: Leah Buechley
 * Date:   8/2021
 * This is an example to help you use the Screen class
 * Refer to Java graphics documentation for information on drawing:
 * https://docs.oracle.com/en/java/javase/16/docs/api/java.desktop/java/awt/Gra
 ***********************************************************************

import java.awt.*;

public class ScreenExample {
    //Create a screen/window to draw in
    static Screen screen= new Screen();

    //Main just paints the screen over and over forever
    public static void main(String[] args) {
        while (true) {
            paint();
        }
    }

    //The paint() method is where all the interesting stuff happens
    public static void paint() {
        //clear the screen
        screen.clearScreen();
        Color backgroundColor = new Color(196, 154, 6);
```

RENAME CLASS Methods

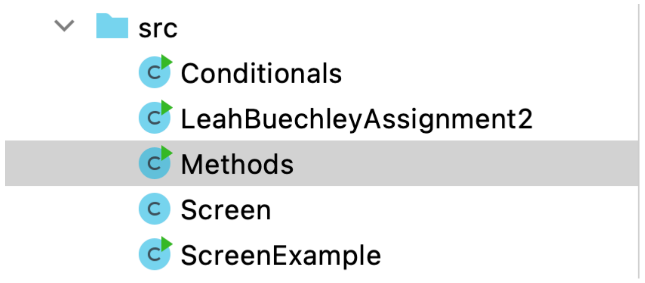


- src
  - Conditionals
  - LeahBuechleyAssignment2
  - Methods
  - Screen
  - ScreenExample

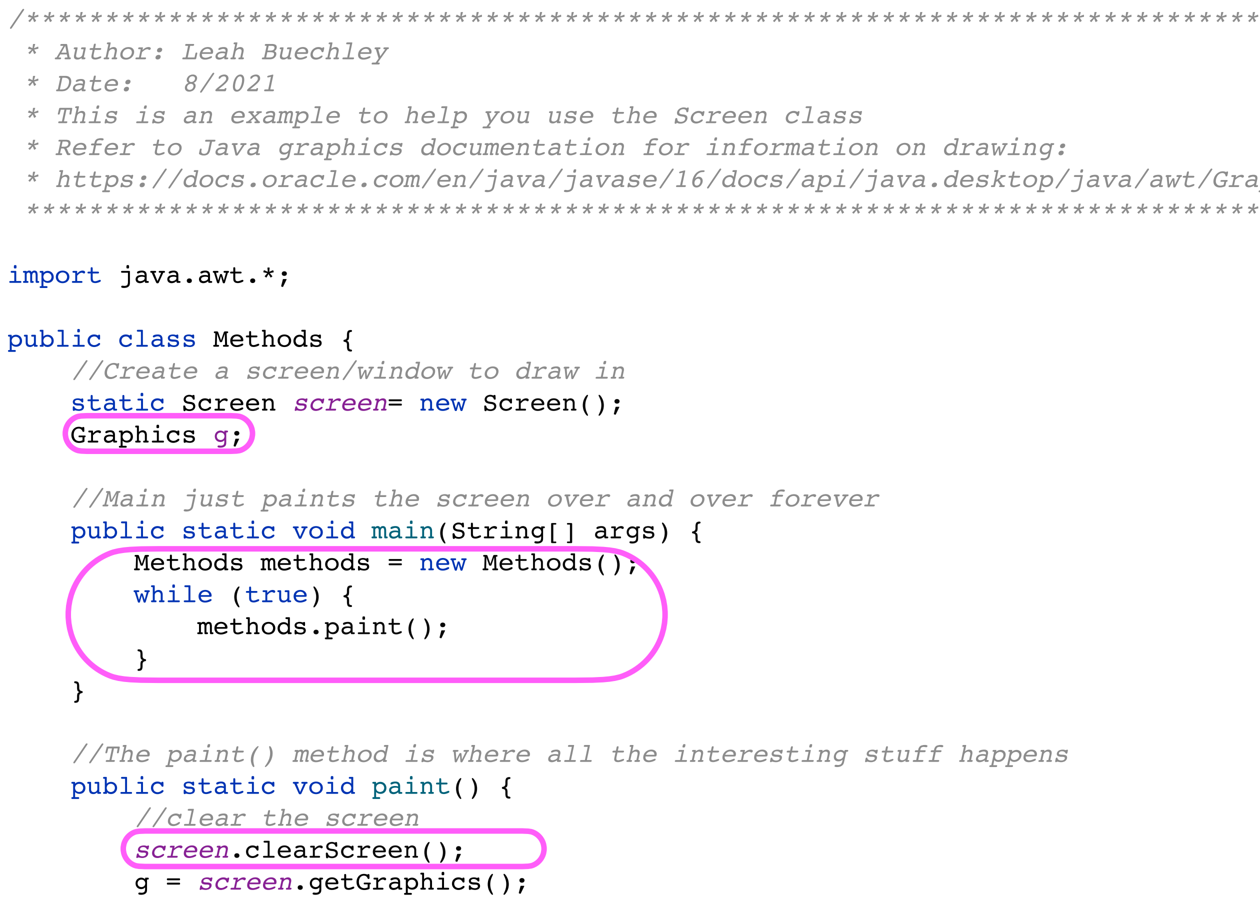


```
/****************************************************************************
 * Author: Leah Buechley
 * Date:   8/2021
 * This is an example to help you use the Screen class
 * Refer to Java graphics documentation for information on drawing:
 * https://docs.oracle.com/en/java/javase/16/docs/api/java.desktop/java/awt/Gra
 ****************************************************************************

import java.awt.*;

public class Methods {
    //Create a screen/window to draw in
    static Screen screen= new Screen();

    //Main just paints the screen over and over forever
    public static void main(String[] args) {
        while (true) {
            paint();
        }
    }

    //The paint() method is where all the interesting stuff happens
    public static void paint() {
        //clear the screen
        screen.clearScreen();
        g = screen.getGraphics();

        //Do all drawing here
```

MAKE A FEW EDITS

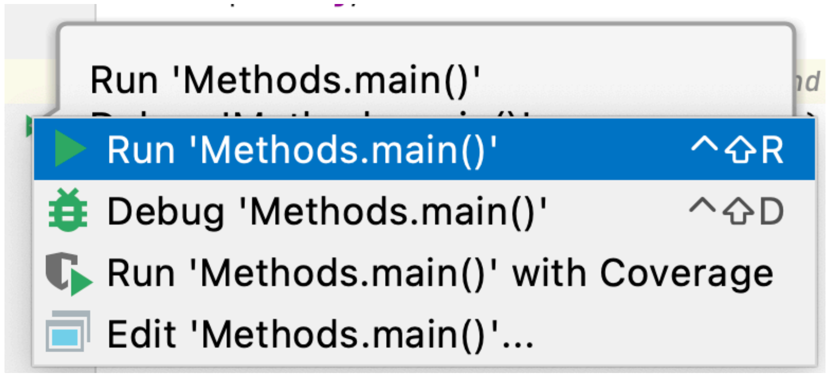


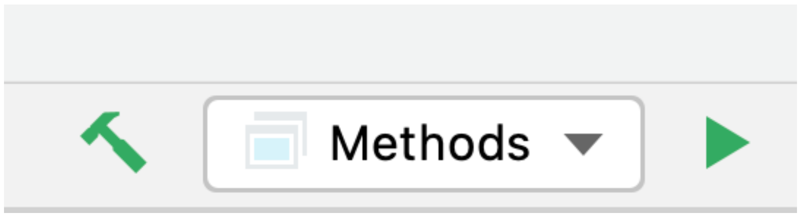
```
src
    Conditionals
    LeahBuechleyAssignment2
    Methods
    Screen
    ScreenExample
```



```
/*****************************************************************************
 * Author: Leah Buechley
 * Date:   8/2021
 * This is an example to help you use the Screen class
 * Refer to Java graphics documentation for information on drawing:
 * https://docs.oracle.com/en/java/javase/16/docs/api/java.desktop/java/awt/Gra
 *****************************************************************************

import java.awt.*;

public class Methods {
    //Create a screen/window to draw in
    static Screen screen= new Screen();
    Graphics g;

    //Main just paints the screen over and over forever
    public static void main(String[] args) {
        Methods methods = new Methods();
        while (true) {
            methods.paint();
        }
    }

    //The paint() method is where all the interesting stuff happens
    public static void paint() {
        //clear the screen
        screen.clearScreen();
        g = screen.getGraphics();
```

RUN. CLICK ON ARROW NEXT TO MAIN SELECT RUN Methods.main()



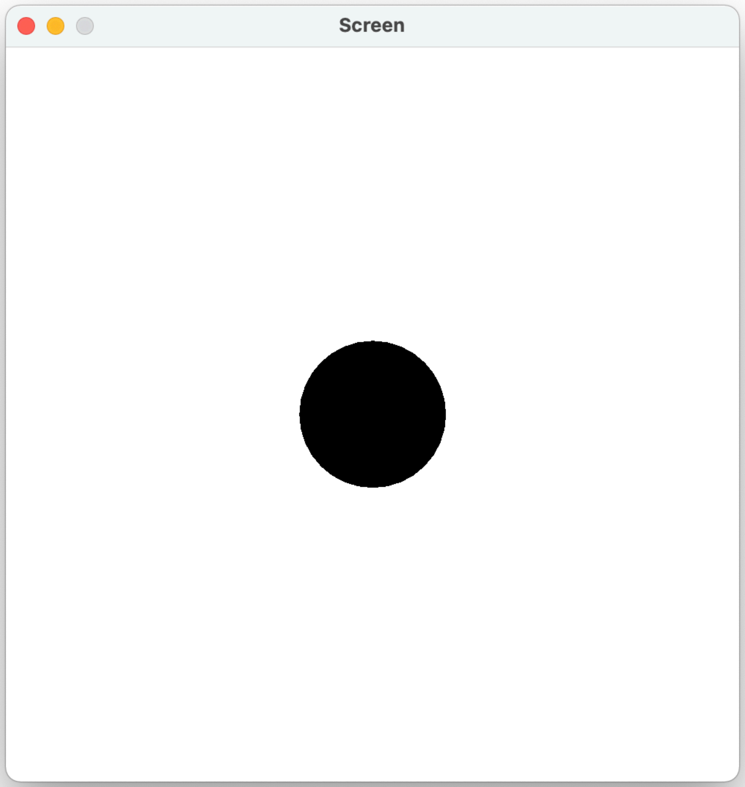


LETS DRAW A FILLED CIRCLE CENTERED ON THE SCREEN

```java
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    g.drawRect(150,150,50,50);

    //update the screen with the drawing that you made
    screen.update(g);
}
```

CENTERED CIRCLE

```java
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    int size = 100;
    g.fillOval(screen.width/2-size/2, screen.height/2-size/2, size, size);

    //update the screen with the drawing that you made
    screen.update(g);
}
```


Screen

PUT IT IN A METHOD AKA FUNCTION

```
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredCircle() {
    int size = 100;
    g.fillOval(screen.width/2-size/2, screen.height/2-size/2, size, size);
}
```

STRUCTURE of METHODS in JAVA

return type
“void” means
nothing is returned

name

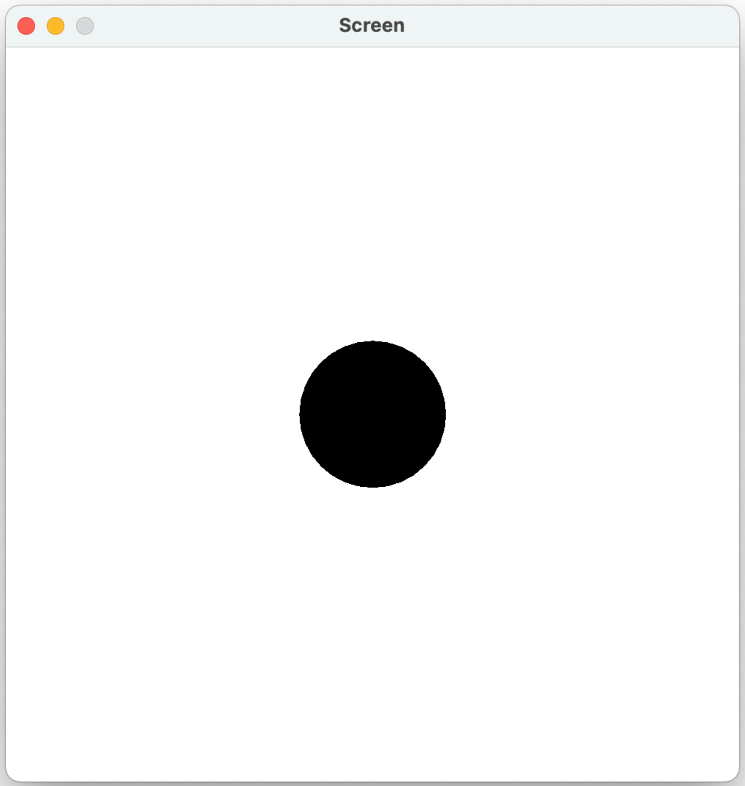
arguments, with their type

```
void fillCenteredCircle(int x, int y, int size) {
    g.fillOval(x-size/2,y-size/2, size, size);
}
```

body of method
inside curly brackets

CALL YOUR METHOD

```
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle();

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredCircle() {
    int size = 100;
    g.fillOval(screen.width/2-size/2, screen.height/2-size/2, size, size);
}
```


Screen

questions?

A METHOD WITH PARAMETERS

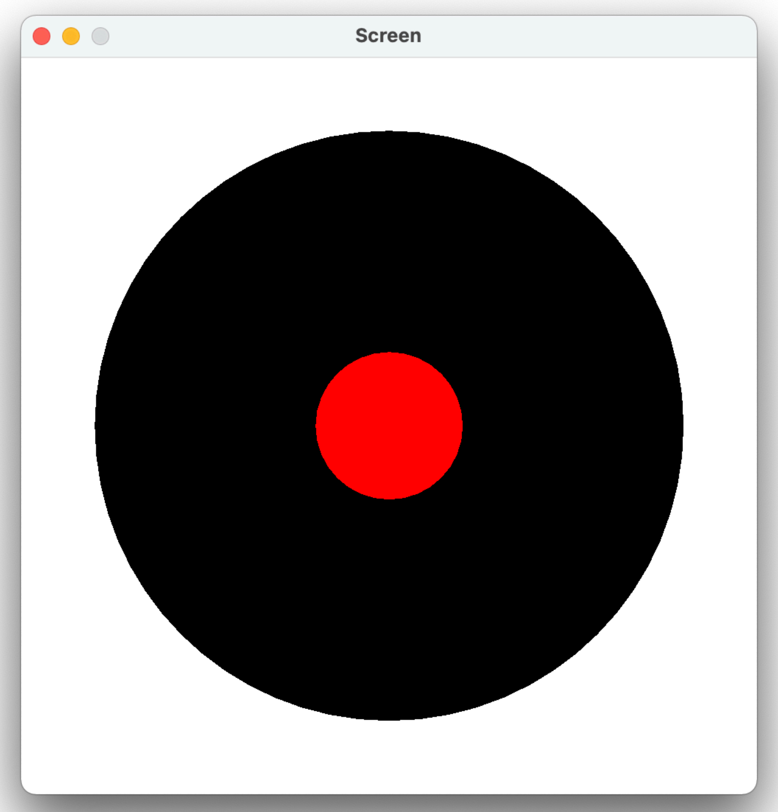
```
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle();

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredCircle() {
    int size = 100;
    g.fillOval(screen.width/2-size/2, screen.height/2-size/2, size, size);
}
```

A METHOD WITH PARAMETERS

```
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle(100);

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredCircle(int size) {
    g.fillOval(screen.width/2-size/2, screen.height/2-size/2, size, size);
}
```

A METHOD WITH PARAMETERS

```java
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle(400);
    g.setColor(Color.RED);
    fillCenteredCircle(100);

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredCircle(int size) {
    g.fillOval(screen.width/2-size/2, screen.height/2-size/2, size, size);
}
```


Screen

OTHER USEFUL PARAMETERS?

X and Y POSITION

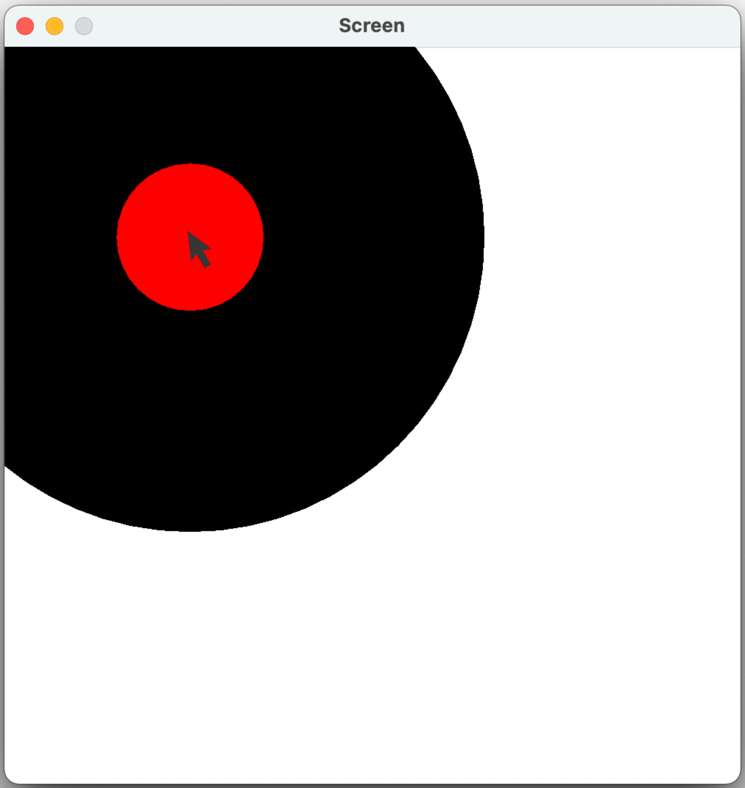
```
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle(400);
    g.setColor(Color.RED);
    fillCenteredCircle(100);

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredCircle(int x, int y, int size) {
    g.fillOval(x-size/2, y-size/2, size, size);
}
```

ADDING POSITION TO THE CALL

```
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle(screen.width/2, screen.height/2, 400);
    g.setColor(Color.RED);
    fillCenteredCircle(screen.width/2, screen.height/2  100);

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredCircle(int x, int y, int size) {
    g.fillOval(x-size/2,y-size/2, size, size);
}
```

CHANGING POSITION

```
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
    g.setColor(Color.RED);
    fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredCircle(int x, int y, int size) {
    g.fillOval(x-size/2,y-size/2, size, size);
}
```


Screen

questions?

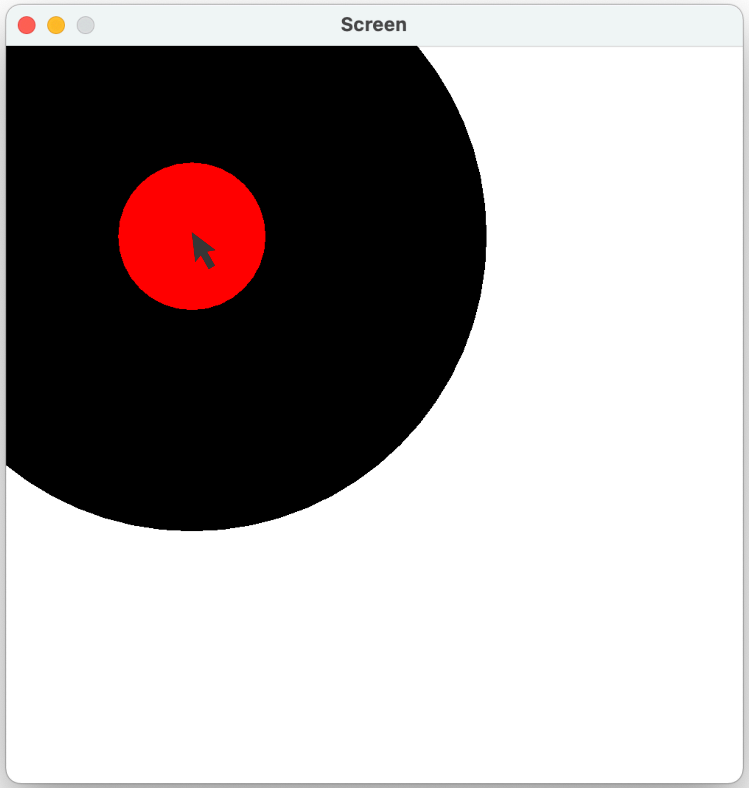
FUNCTIONS/METHODS

WHAT ARE METHODS?

- A chunk of code that you give a name to.
- You “call” a function by writing it’s name in your program
- When your program encounters the name, it jumps to the function and executes it.
- When finished, the program “returns” to where it was when the function was called.

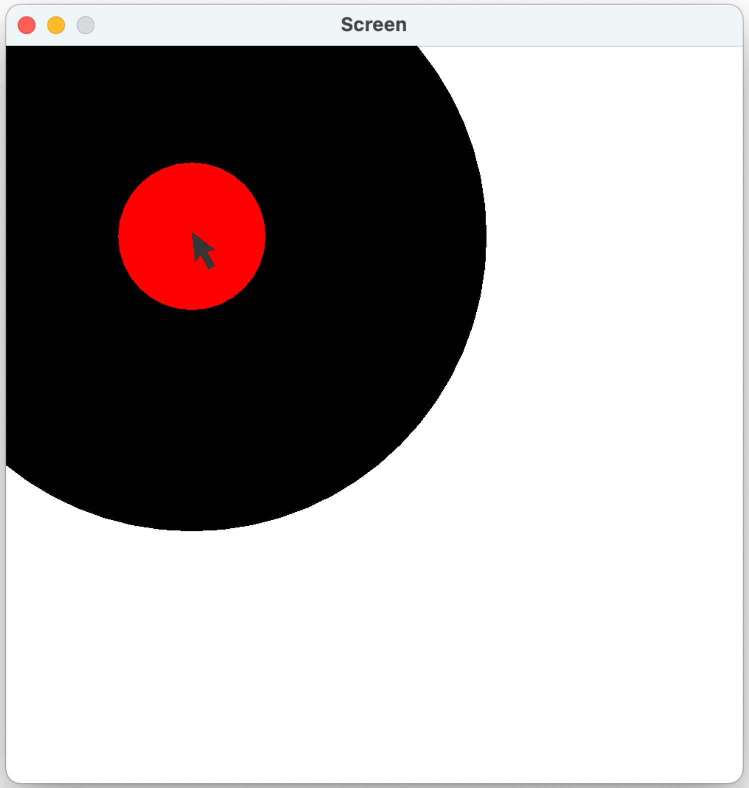
HOW METHODS WORK

```java
public static void paint() {
     //clear the screen
     screen.clearScreen();
     g = screen.getGraphics();

     //Do all drawing here
     g.setColor(Color.BLACK);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
     g.setColor(Color.RED);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

     //update the screen with the drawing that you made
     screen.update(g);
 }

 void fillCenteredCircle(int x, int y, int size) {
     g.fillOval(x-size/2,y-size/2, size, size);
 }
```



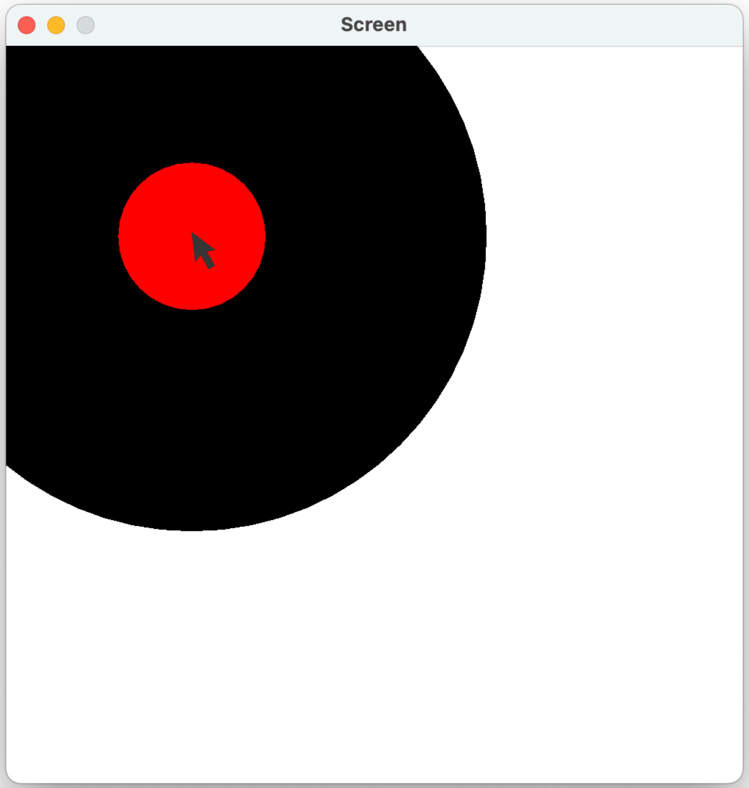
HOW METHODS WORK

```
public static void paint() {
     //clear the screen
     screen.clearScreen();
     g = screen.getGraphics();

     //Do all drawing here
     g.setColor(Color.BLACK);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
     g.setColor(Color.RED);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

     //update the screen with the drawing that you made
     screen.update(g);
 }

 void fillCenteredCircle(int x, int y, int size) {
     g.fillOval(x-size/2,y-size/2, size, size);
 }
```



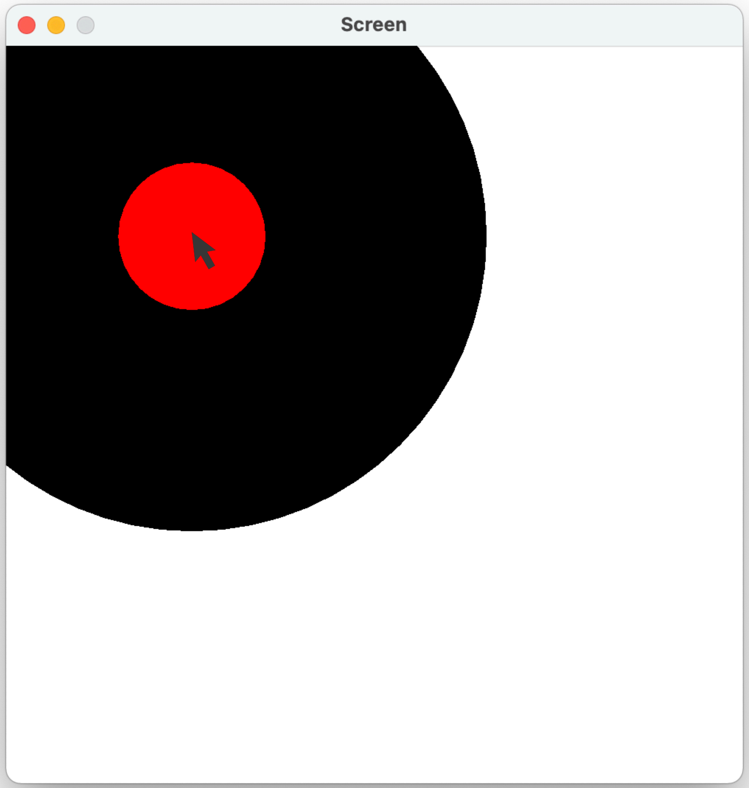
HOW METHODS WORK

```
public static void paint() {
     //clear the screen
     screen.clearScreen();
     g = screen.getGraphics();

     //Do all drawing here
     g.setColor(Color.BLACK);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
     g.setColor(Color.RED);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

     //update the screen with the drawing that you made
     screen.update(g);
 }

 void fillCenteredCircle(int x, int y, int size) {
     g.fillOval(x-size/2,y-size/2, size, size);
 }
```



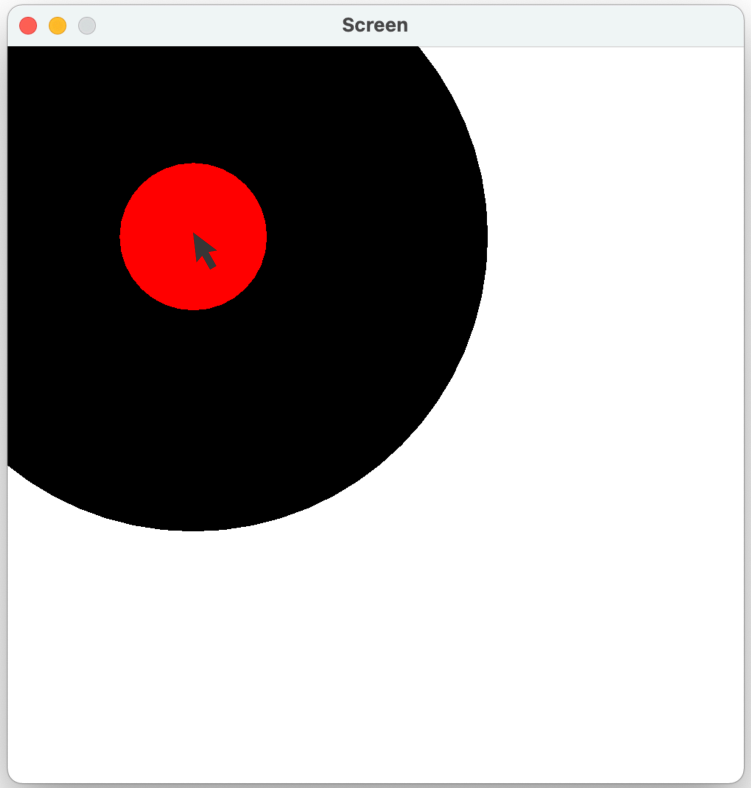
HOW METHODS WORK

```
public static void paint() {
     //clear the screen
     screen.clearScreen();
     g = screen.getGraphics();

     //Do all drawing here
     g.setColor(Color.BLACK);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
     g.setColor(Color.RED);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

     //update the screen with the drawing that you made
     screen.update(g);
 }

 void fillCenteredCircle(int x, int y, int size) {
     g.fillOval(x-size/2,y-size/2, size, size);
 }
```



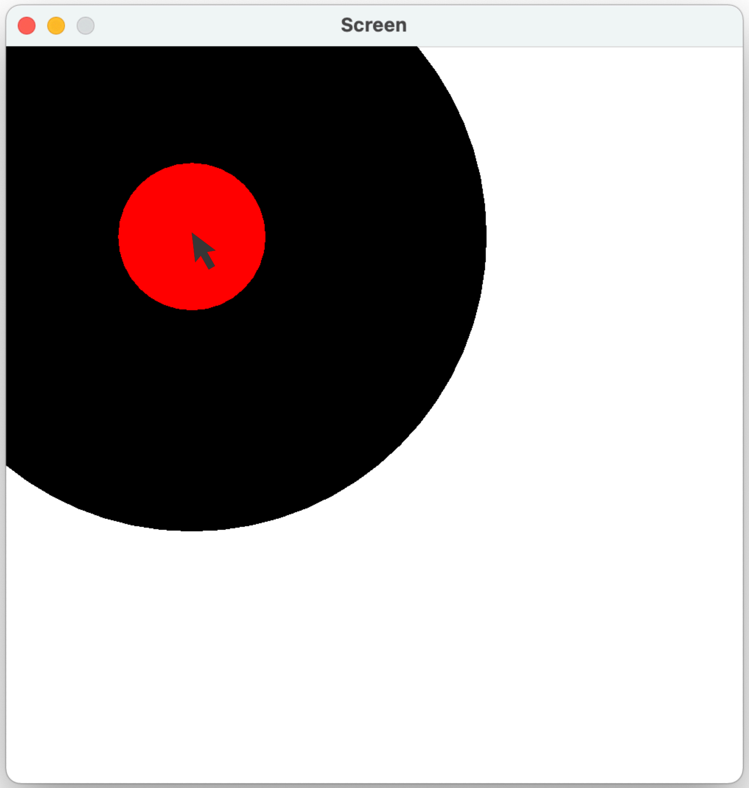
HOW METHODS WORK

```
public static void paint() {
     //clear the screen
     screen.clearScreen();
     g = screen.getGraphics();

     //Do all drawing here
     g.setColor(Color.BLACK);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
     g.setColor(Color.RED);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

     //update the screen with the drawing that you made
     screen.update(g);
 }

 void fillCenteredCircle(int x, int y, int size) {
     g.fillOval(x-size/2,y-size/2, size, size);
 }
```



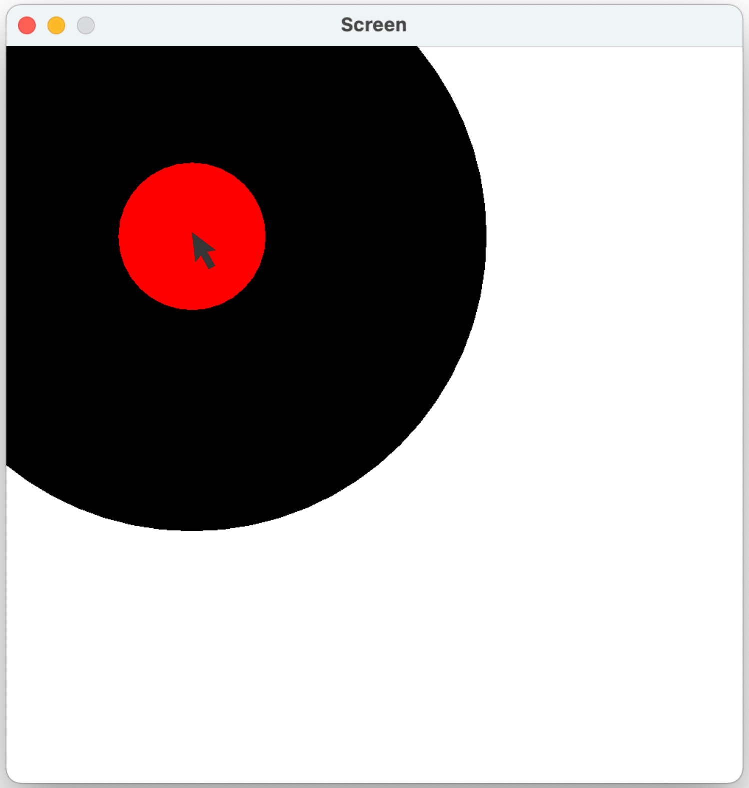
HOW METHODS WORK

```
public static void paint() {
     //clear the screen
     screen.clearScreen();
     g = screen.getGraphics();

     //Do all drawing here
     g.setColor(Color.BLACK);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
     g.setColor(Color.RED);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

     //update the screen with the drawing that you made
     screen.update(g);
 }

 void fillCenteredCircle(int x, int y, int size) {
     g.fillOval(x-size/2,y-size/2, size, size);
 }
```



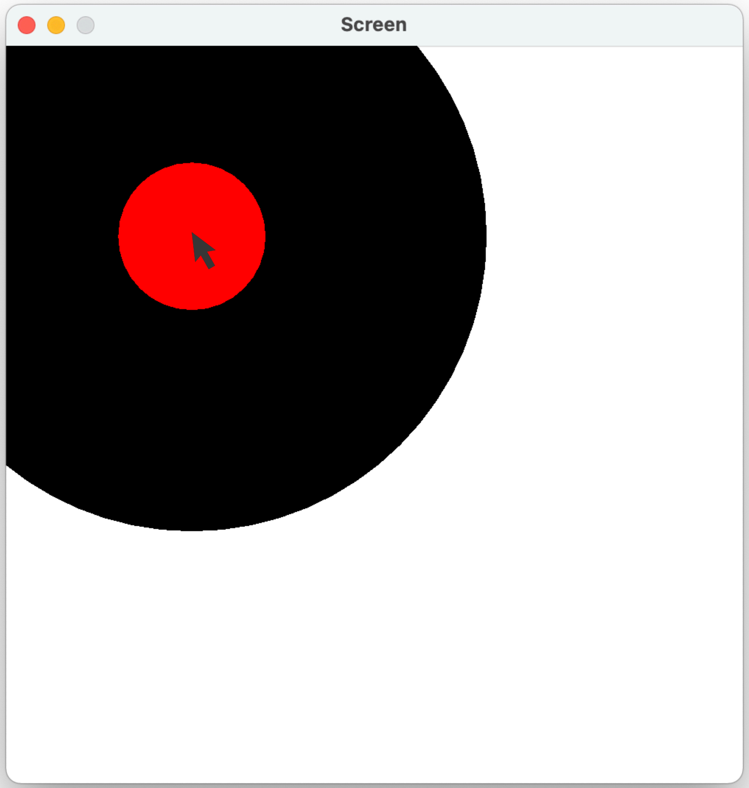
HOW METHODS WORK

```
public static void paint() {
     //clear the screen
     screen.clearScreen();
     g = screen.getGraphics();

     //Do all drawing here
     g.setColor(Color.BLACK);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
     g.setColor(Color.RED);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

     //update the screen with the drawing that you made
     screen.update(g);
 }

 void fillCenteredCircle(int x, int y, int size) {
     g.fillOval(x-size/2,y-size/2, size, size);
 }
```



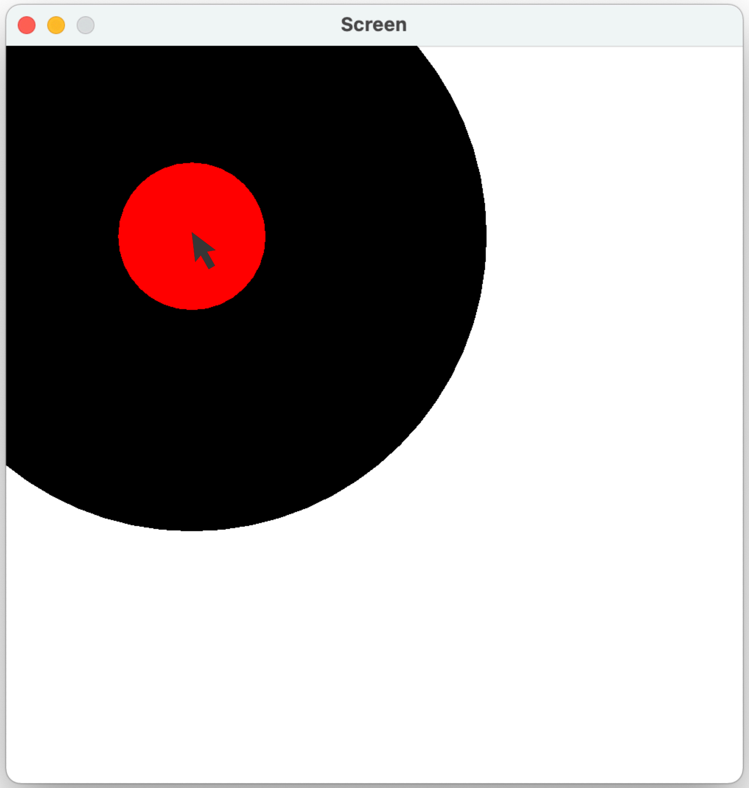
HOW METHODS WORK

```
public static void paint() {
     //clear the screen
     screen.clearScreen();
     g = screen.getGraphics();

     //Do all drawing here
     g.setColor(Color.BLACK);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
     g.setColor(Color.RED);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

     //update the screen with the drawing that you made
     screen.update(g);
 }

 void fillCenteredCircle(int x, int y, int size) {
     g.fillOval(x-size/2,y-size/2, size, size);
 }
```



HOW METHODS WORK

```
public static void paint() {
     //clear the screen
     screen.clearScreen();
     g = screen.getGraphics();

     //Do all drawing here
     g.setColor(Color.BLACK);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
     g.setColor(Color.RED);
     fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

     //update the screen with the drawing that you made
     screen.update(g);
 }

 void fillCenteredCircle(int x, int y, int size) {
     g.fillOval(x-size/2,y-size/2, size, size);
 }
```



STRUCTURE of METHODS in JAVA

return type
“void” means
nothing is returned

name

arguments, with their type

```
void fillCenteredCircle(int x, int y, int size) {
    g.fillOval(x-size/2,y-size/2, size, size);
}
```

body of method
inside curly brackets

A FUNCTION THAT RETURNS A VALUE

return type

```
int addTwoNumbers(int num1, int num2) {
    int result;
    result = num1 + num2;
    return result;
}
```

return statement
must be present
value type must match return type

ANOTHER METHOD

```
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
    g.setColor(Color.RED);
    fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredOval(int x, int y, int width, int height) {
    g.fillOval(x-width/2, y-height/2, width, height);
}

void fillCenteredCircle(int x, int y, int size) {
    g.fillOval(x-size/2,y-size/2, size, size);
}
```

AND ANOTHER

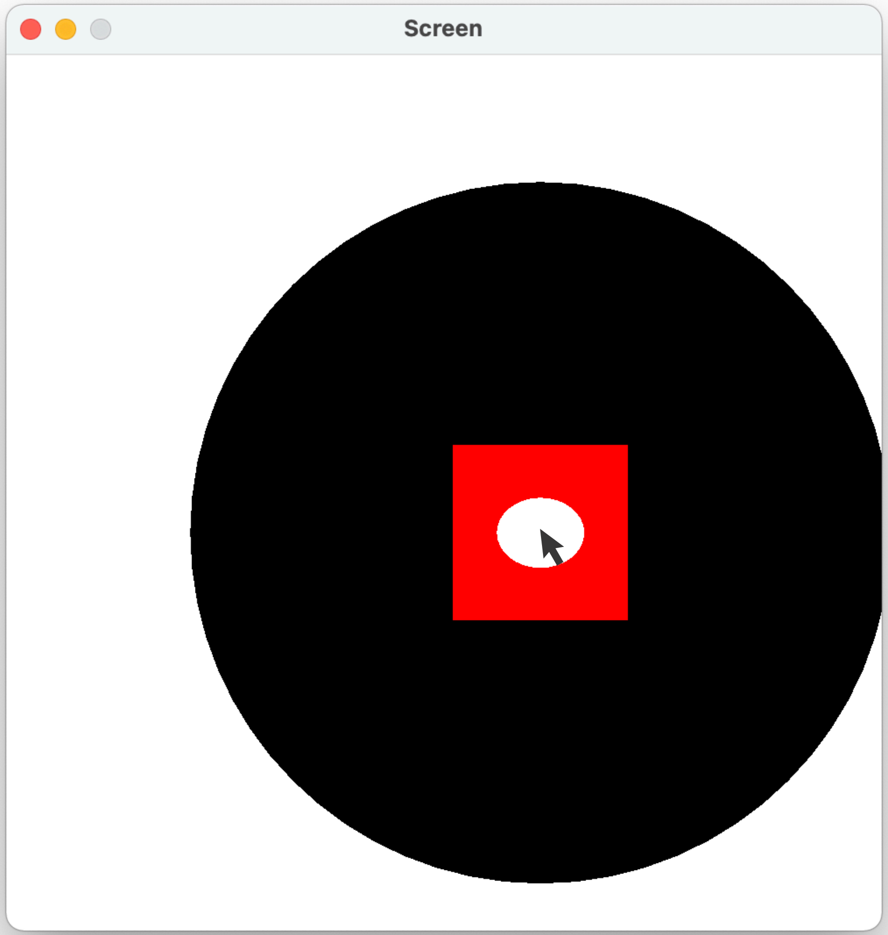
```java
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
    g.setColor(Color.RED);
    fillCenteredCircle(screen.mouseX, screen.mouseY, 100);

    //update the screen with the drawing that you made
    screen.update(g);
}

void fillCenteredOval(int x, int y, int width, int height) {
    g.fillOval(x-width/2, y-height/2, width, height);
}

void fillCenteredRect(int x, int y, int width, int height) {
    g.fillRect(x-width/2, y-height/2, width, height);
}

void fillCenteredCircle(int x, int y, int size) {
    g.fillOval(x-size/2,y-size/2, size, size);
}
```

ONE LAST ONE

```
void fillCenteredOval(int x, int y, int width, int height) {
    g.fillOval(x-width/2, y-height/2, width, height);
}

void fillCenteredRect(int x, int y, int width, int height) {
    g.fillRect(x-width/2, y-height/2, width, height);
}

void fillCenteredCircle(int x, int y, int size) {
    g.fillOval(x-size/2,y-size/2, size, size);
}

void fillCenteredSquare(int x, int y, int size) {
    g.fillRect(x-size/2,y-size/2, size, size);
}
```

COMBINING THEM

```
//The paint() method is where all the interesting stuff happens
public static void paint() {
    //clear the screen
    screen.clearScreen();
    g = screen.getGraphics();

    //Do all drawing here
    g.setColor(Color.BLACK);
    fillCenteredCircle(screen.mouseX, screen.mouseY, 400);
    g.setColor(Color.RED);
    fillCenteredSquare(screen.mouseX, screen.mouseY, 100);
    g.setColor(Color.WHITE);
    fillCenteredOval(screen.mouseX, screen.mouseY, 50, 40);

    //update the screen with the drawing that you made
    screen.update(g);
}
```


Screen

questions?

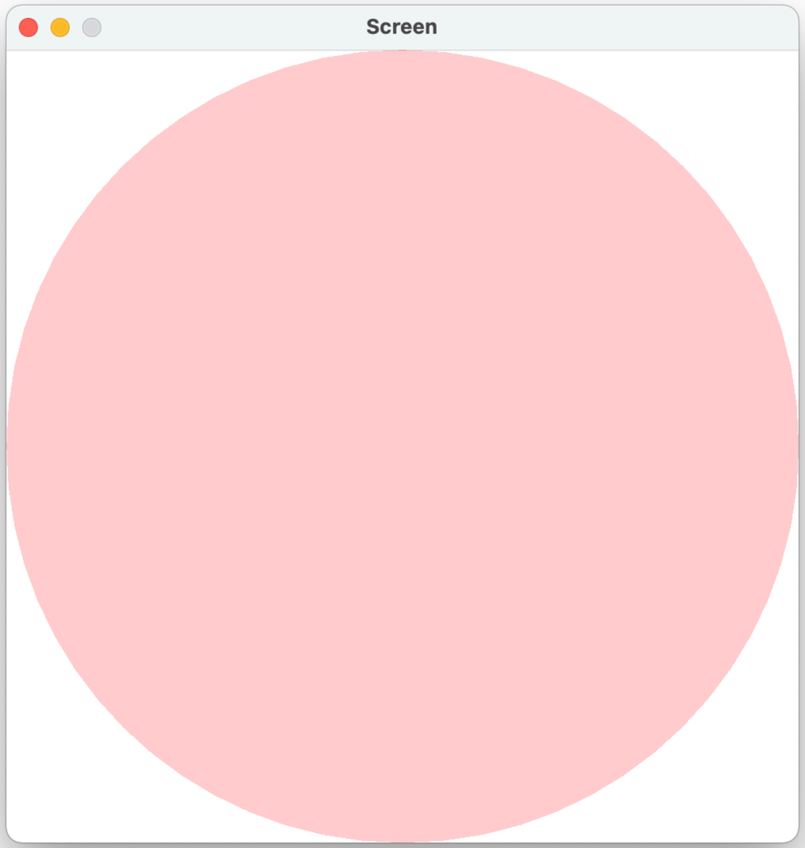
ANOTHER NEW METHOD

```
void transparentCircle () {
    int transparency = 50;
    Color circleColor = new Color(255, 0, 0, transparency);
    g.setColor(circleColor);
    fillCenteredCircle(screen.width/2, screen.height/2, screen.width);
}
```

ANOTHER NEW METHOD

```
void transparentCircle () {
    int transparency = 50;
    Color circleColor = new Color(255, 0, 0, transparency);
    g.setColor(circleColor);
    fillCenteredCircle(screen.width/2, screen.height/2, screen.width);
}
```

can call other methods we’ve defined


Screen

COLOR & TRANSPARENCY

```
Color circleColor = new Color(255, 0, 0, transparency);
```

sets the transparency or "alpha"
of the color. ranges from:
0 = clear
255 = opaque

PLAY WITH TRANSPARENCY VALUES

```
void transparentCircle () {
    int transparency = 50;
    Color circleColor = new Color(255, 0, 0, transparency);
    g.setColor(circleColor);
    fillCenteredCircle(screen.width/2, screen.height/2, screen.width);
}
```

PLAY WITH TRANSPARENCY VALUES

```
void transparentCircle () {
    int transparency = 200;
    Color circleColor = new Color(255, 0, 0, transparency);
    g.setColor(circleColor);
    fillCenteredCircle(screen.width/2, screen.height/2, screen.width);
}
```

PLAY WITH TRANSPARENCY VALUES

```
void transparentCircle () {
    int transparency = 10;
    Color circleColor = new Color(255, 0, 0, transparency);
    g.setColor(circleColor);
    fillCenteredCircle(screen.width/2, screen.height/2, screen.width);
}
```

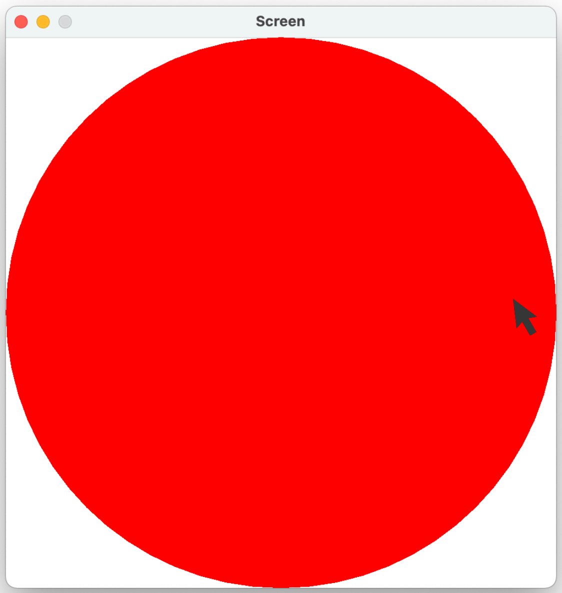
PLAY WITH TRANSPARENCY VALUES

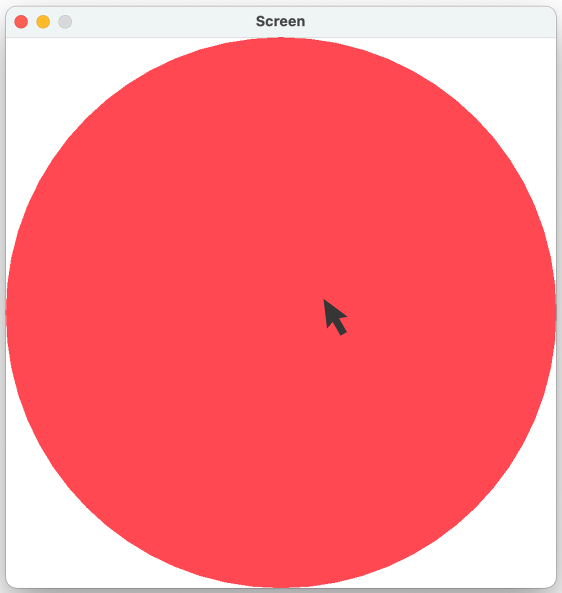
will range from 0 to 250
for a 500 pixel screen

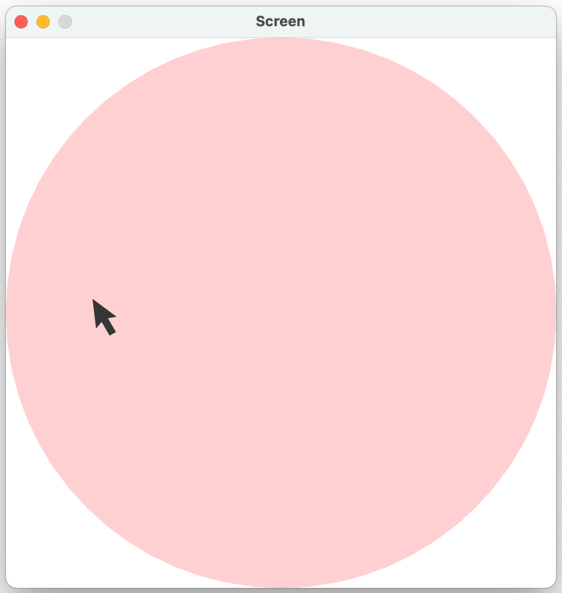
```
void transparentCircle () {
    int transparency = screen.mouseX/2;
    Color circleColor = new Color(255, 0, 0, transparency);
    g.setColor(circleColor);
    fillCenteredCircle(screen.width/2, screen.height/2, screen.width);
}
```

PRINT VALUES

```
void transparentCircle () {
    int transparency = screen.mouseX/2;
    System.out.println(transparency);
    Color circleColor = new Color(255, 0, 0, transparency);
    g.setColor(circleColor);
    fillCenteredCircle(screen.width/2, screen.height/2, screen.width);
}
```


Screen


Screen


Screen

questions?

ESS PRESENTATION

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/