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/

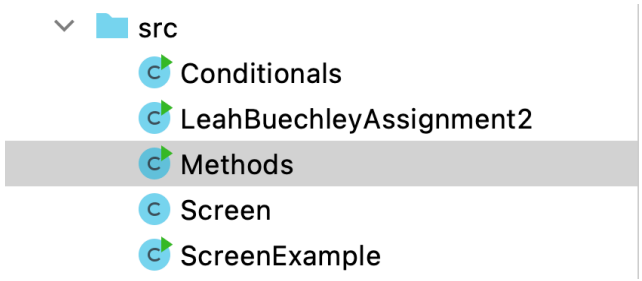
UNM DROP DEADLINE

- Today
- Last chance to drop without paying tuition
- Will receive W on transcript

OPEN INTELLIJ &
LAST WEEK'S PROJECT

CORRECTION FROM LAST CLASS
CODE POSTED ONLINE

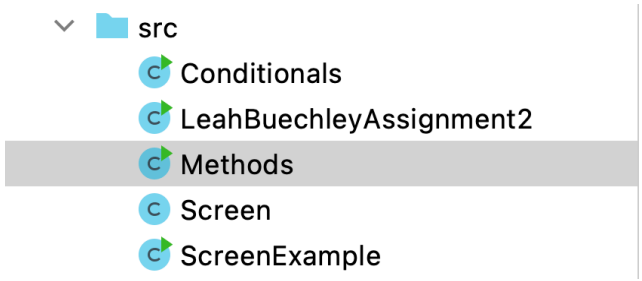
Methods.java




```java
/*******************************************************************************
 * 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();
```

COPY AND PASTE CODE ONLINE INTO YOUR Methods.java FILE

Methods.java




```java
/*******************************************************************************
 * 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
PUT AWAY LAPTOPS FOR A BIT

CODE STRUCTURE

CLASS {

}

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

        //Do all drawing here
        transparentCircle(screen.width/2, screen.height/2, 300);

        //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);
    }

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

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

CLASS {

Variables

}

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

        //Do all drawing here
        transparentCircle(screen.width/2, screen.height/2, 300);

        //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);
    }

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

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

CLASS {

Variables

main {

}

}

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

        //Do all drawing here
        transparentCircle(screen.width/2, screen.height/2, 300);

        //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);
    }

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

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

CLASS {

Variables

main {
}

paint {
}

}

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

        //Do all drawing here
        transparentCircle(screen.width/2, screen.height/2, 300);

        //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);
    }

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

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

CLASS {

Variables

main {
}

paint {
}

method1 {
}

}

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

        //Do all drawing here
        transparentCircle(screen.width/2, screen.height/2, 300);

        //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);
    }

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

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

```
CLASS {
  Variables

  main {
  }

  paint {
  }

  method1 {
  }

  method2 {
  }
}
```

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

        //Do all drawing here
        transparentCircle(screen.width/2, screen.height/2, 300);

        //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);
    }

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

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

CLASS {

Variables

main {
}

paint {
}

method1 {
}

method2 {
}

...

}

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

        //Do all drawing here
        transparentCircle(screen.width/2, screen.height/2, 300);

        //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);
    }

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

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

STRUCTURE PROBLEMS

STRUCTURE PROBLEMS
MISSING CURLY BRACKETS

```
CLASS {
  Variables

  main {
  }

  paint {
  }

  method1 {
  }

  method2 {
  }
  ...
}
```

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

    //Do all drawing here
    transparentCircle(screen.width/2, screen.height/2, 300);

    //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);
    }

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

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

CLASS {

Variables

main {
}

paint {

method1 {
}

method2 {
}

...

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

    //Do all drawing here
    transparentCircle(screen.width/2, screen.height/2, 300);

    //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);
    }

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

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

MISSING CURLY BRACKETS = UNHELPFUL COMPILE ERRORS

```
/Users/LAB 1/websites/CS152_Fall2021/javaCode/Wed_09_01/src/Methods.java:37:5
java: illegal start of expression
```

```
/Users/LAB 1/websites/CS152_Fall2021/javaCode/Wed_09_01/src/Methods.java:50:6
java: reached end of file while parsing
```

```
/Users/LAB 1/websites/CS152_Fall2021/javaCode/Wed_09_01/src/Methods.java:41:52
java: ';' expected
```

STRUCTURE PROBLEMS
EXTRA CURLY BRACKETS

```
CLASS {
  Variables

  main {
  }

  paint {
  }

  method1 {
  }

  method2 {
  }
  ...
}
```

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

    //Do all drawing here
    transparentCircle(screen.width/2, screen.height/2, 300);

    //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);
    }

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

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

```
CLASS {
  Variables

  main {
  }

  paint {
  }

  method1 {
  }}

  method2 {
  }
  ...
}
```

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

    //Do all drawing here
    transparentCircle(screen.width/2, screen.height/2, 300);

    //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);
    }}

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

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

EXTRA CURLY BRACKETS = UNHELPFUL COMPILE ERRORS

```
/Users/LAB 1/websites/CS152_Fall2021/javaCode/Wed_09_01/src/Methods.java:51
java: class, interface, enum, or record expected

/Users/LAB 1/websites/CS152_Fall2021/javaCode/Wed_09_01/src/Methods.java:51:2
java: reached end of file while parsing
```

STRUCTURE PROBLEMS
NESTED METHODS

```
CLASS {
  Variables

  main {
  }

  paint {
  }

  method1 {
  }

  method2 {
  }
  ...
}
```

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

    //Do all drawing here
    transparentCircle(screen.width/2, screen.height/2, 300);

    //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);
    }

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

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

```
CLASS {
  Variables

  main {
  }

  paint {

    method1 {
    }

    method2 {
    }

  }
}
```

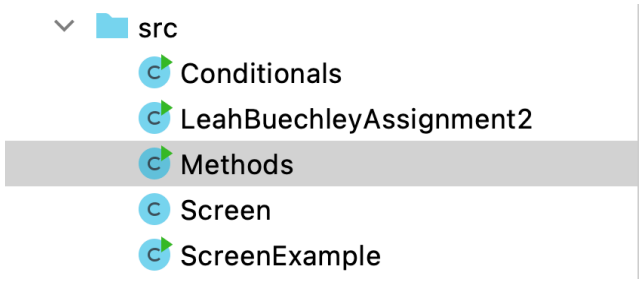
```
import java.awt.*;

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

    //Do all drawing here
    transparentCircle(screen.width/2, screen.height/2, 300);

    //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);
    }

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


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

questions?

IF YOU'RE STUCK
LOOK AT HIGH LEVEL STRUCTURE

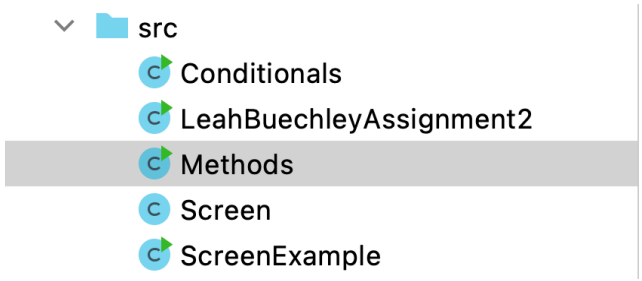
BACK TO CODING

Methods.java




```java
/*******************************************************************************
 * 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();
```

Methods.java




```java
/*******************************************************************************
 * 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();
```

ADD A NEW METHOD TO OUR CODE

ADD A NEW METHOD TO OUR CODE CALLED circleLine

ADD A NEW METHOD TO OUR CODE CALLED circleLine RIGHT BELOW paint()

CLASS {

Variables

main {
}

paint {
}

circleLine {
}

method1 {
}

...

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

    //Do all drawing here
    transparentCircle(screen.width/2, screen.height/2, 300);

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

    void circleLine() {

    }

    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);
    }

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

circleLine METHOD

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

//Do all drawing here
transparentCircle(screen.width/2, screen.height/2, 300);

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

void circleLine() {

}
```

IT WILL DRAW A LINE OF CIRCLES

circleLine METHOD

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

    //Do all drawing here
    transparentCircle(screen.width/2, screen.height/2, 300);

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

void circleLine() {

}
```

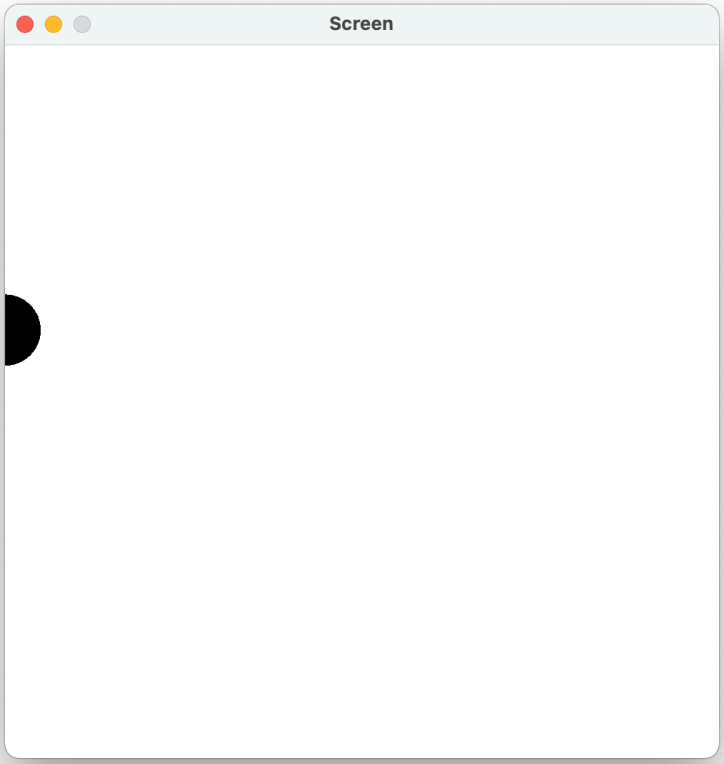
ONE CENTERED CIRCLE at (0,200)

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

    //Do all drawing here
    transparentCircle(screen.width/2, screen.height/2, 300);

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

void circleLine() {
    fillCenteredCircle(0,200,50);
}
```

CALL THE METHOD

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

    //Do all drawing here
    circleLine();

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

void circleLine() {
    fillCenteredCircle(0,200,50);
}
```

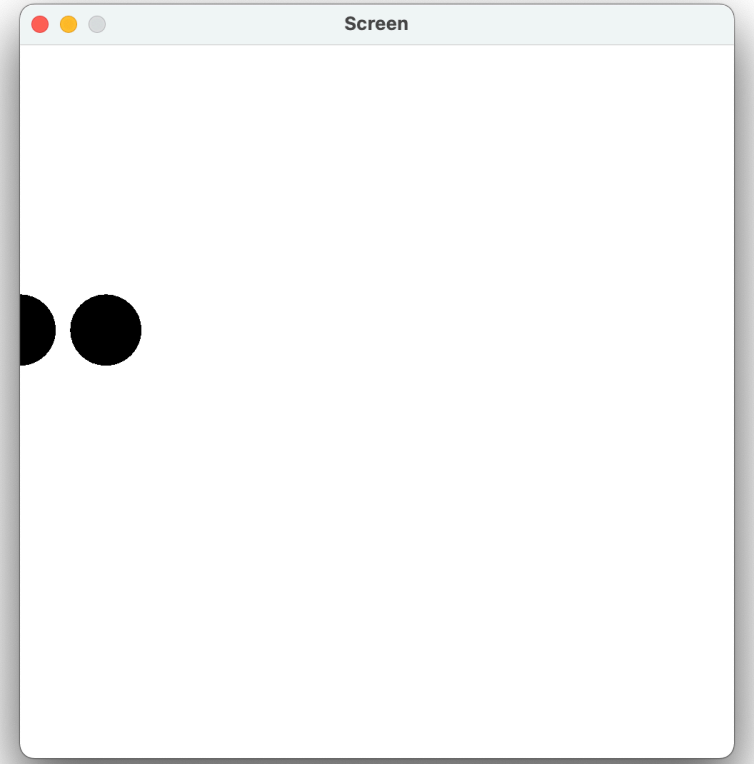
CALL THE METHOD



PICK A NICE COLOR IF YOU WANT

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

    //Do all drawing here
    Color c = new Color(74, 105, 12);
    g.setColor(c);
    circleLine();

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

void circleLine() {
    fillCenteredCircle(0,200,50);
}
```

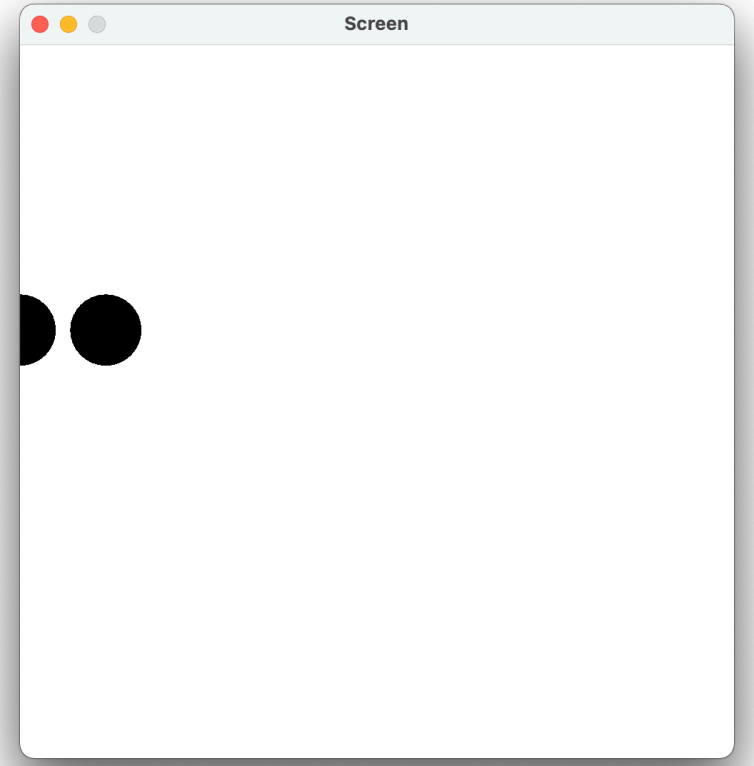
ADD A CIRCLE, 10 PIXELS AWAY IN X



ADD A CIRCLE, 10 PIXELS AWAY IN X

What are the center coordinates of my new circle?

(60, 200)
(0+50+10, 200)



ADD A CIRCLE, 10 PIXELS AWAY IN X

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

    //Do all drawing here
    Color c = new Color(74, 105, 12);
    g.setColor(c);
    circleLine();

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

void circleLine() {
    fillCenteredCircle(0,200,50);
    fillCenteredCircle(50+10,200,50);
}
```

ADD ANOTHER CIRCLE

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

    //Do all drawing here
    Color c = new Color(74, 105, 12);
    g.setColor(c);
    circleLine();

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

void circleLine() {
    fillCenteredCircle(0,200,50);
    fillCenteredCircle(50+10,200,50);
    fillCenteredCircle(50+10+50+10,200,50);
}
```

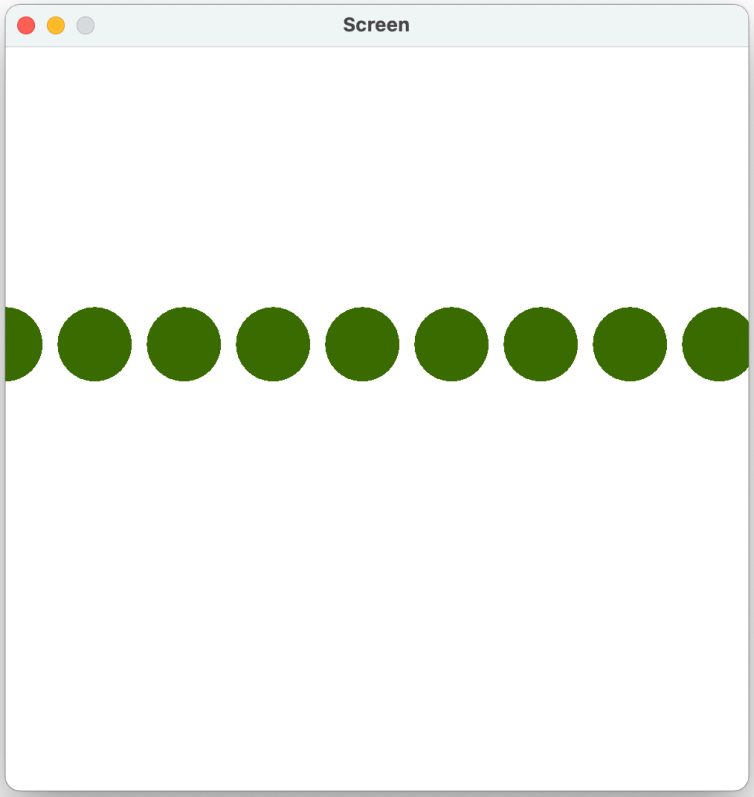
PATTERN OF POSITION

```
fillCenteredCircle(0,200,50);
fillCenteredCircle(50+10,200,50);
fillCenteredCircle(50+10+50+10,200,50);
fillCenteredCircle(50+10+50+10+50+10,200,50);
```

```
newPosition = oldPosition + 50 + 10
x = x + 50 + 10
```

USE A LOOP!

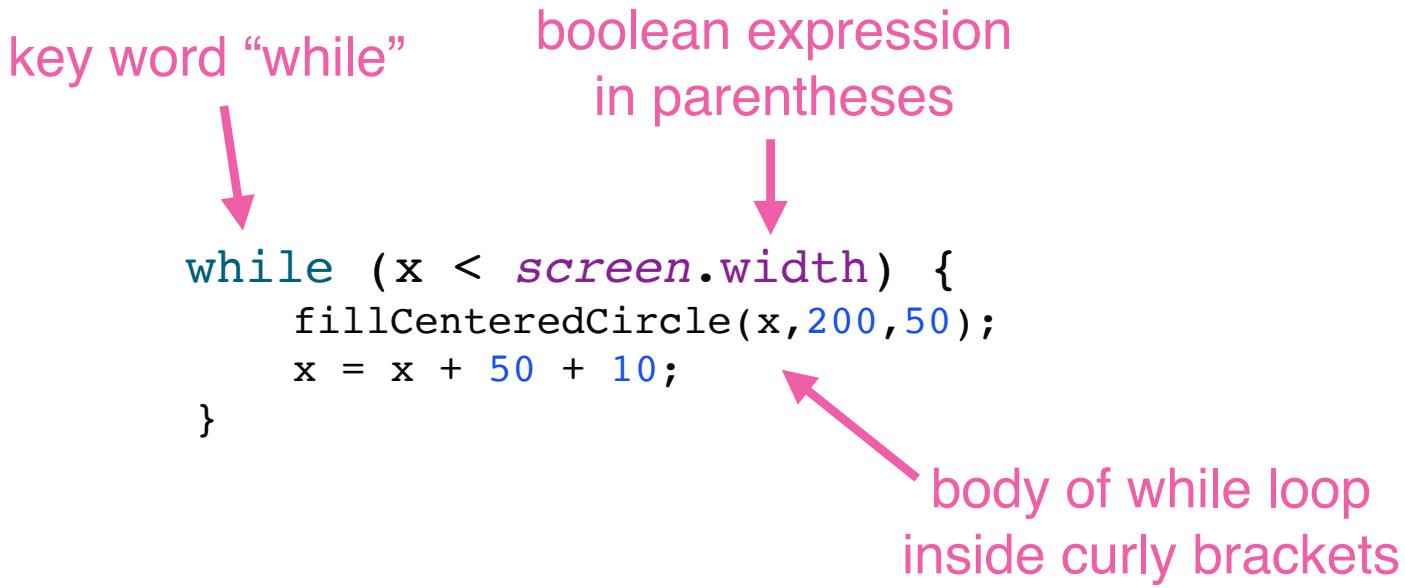
```
void circleLine() {
    int x = 0;
    while (x < screen.width) {
        fillCenteredCircle(x,200,50);
        x = x + 50 + 10;
    }
}
```


Screen

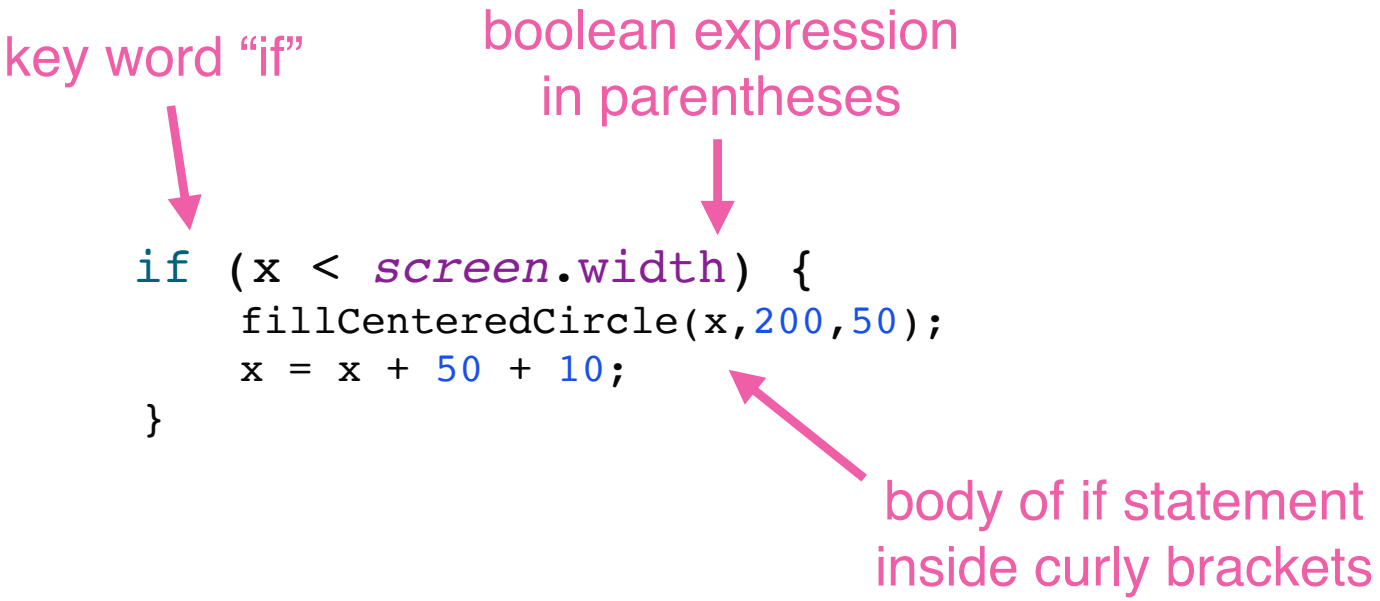
WHILE LOOPS

```
void circleLine() {
    int x = 0;
    while (x < screen.width) {
        fillCenteredCircle(x,200,50);
        x = x + 50 + 10;
    }
}
```

STRUCTURE of WHILE LOOP in JAVA



SIMILAR TO IF



BOOLEAN EXPRESSIONS

- A question with an answer that is either TRUE or FALSE
- A logical statement that is either TRUE or FALSE
- In our program: `screen.mouseX < screen.width/2`
  Is the mouse in the left half of the screen?
- More examples:

  `screen.mouseY < screen.height/2`

  Is the mouse in the upper half of the screen?

  `x <= 50`

  Is x less than or equal to 50?

  `a == 10.5`

  Is a equal to 10.5?

HOW A WHILE LOOP WORKS

while the boolean expression is true

```
while (x < screen.width) {
    fillCenteredCircle(x,200,50);
    x = x + 50 + 10;
}
```

these statements will be executed

DON’T PUT SEMICOLONS HERE!

```
while (x < screen.width); {
    fillCenteredCircle(x,200,50);
    x = x + 50 + 10;
}
```

DON’T PUT SEMICOLONS HERE!

```
while (x < screen.width); {
    fillCenteredCircle(x,200,50);
    x = x + 50 + 10;
}
```

OR HERE!

```
if (x < screen.width); {
    fillCenteredCircle(x,200,50);
    x = x + 50 + 10;
}
```

CORRECT FORM

```
while (x < screen.width) {
    fillCenteredCircle(x,200,50);
    x = x + 50 + 10;
}

if (x < screen.width) {
    fillCenteredCircle(x,200,50);
    x = x + 50 + 10;
}
```

questions?

WHILE LOOPS CAN BE DANGEROUS...

```
void circleLine() {
    int x = 0;
    while (x < screen.width) {
        fillCenteredCircle(x,200,50);
        x = x + 50 + 10;
    }
}
```

INFINTE LOOPS!

this statement is always true
because x doesn't change in loop

```
int x = 0;
while (x < screen.width) {
     fillCenteredCircle(x,200,50);
}
```

RUN THIS CODE & SEE WHAT HAPPENS

```
void circleLine() {
    int x = 0;
    while (x < screen.width) {
        fillCenteredCircle(x,200,50);
        //x = x + 50 + 10;
    }
}
```

BE CAREFUL ABOUT WHILE LOOP CONDITIONS

HOW CAN WE MAKE OUR METHOD BETTER?

```
void circleLine() {
    int x = 0;
    while (x < screen.width) {
        fillCenteredCircle(x, 200, size);
        x = x+size+space;
    }
}
```

VARIABLES

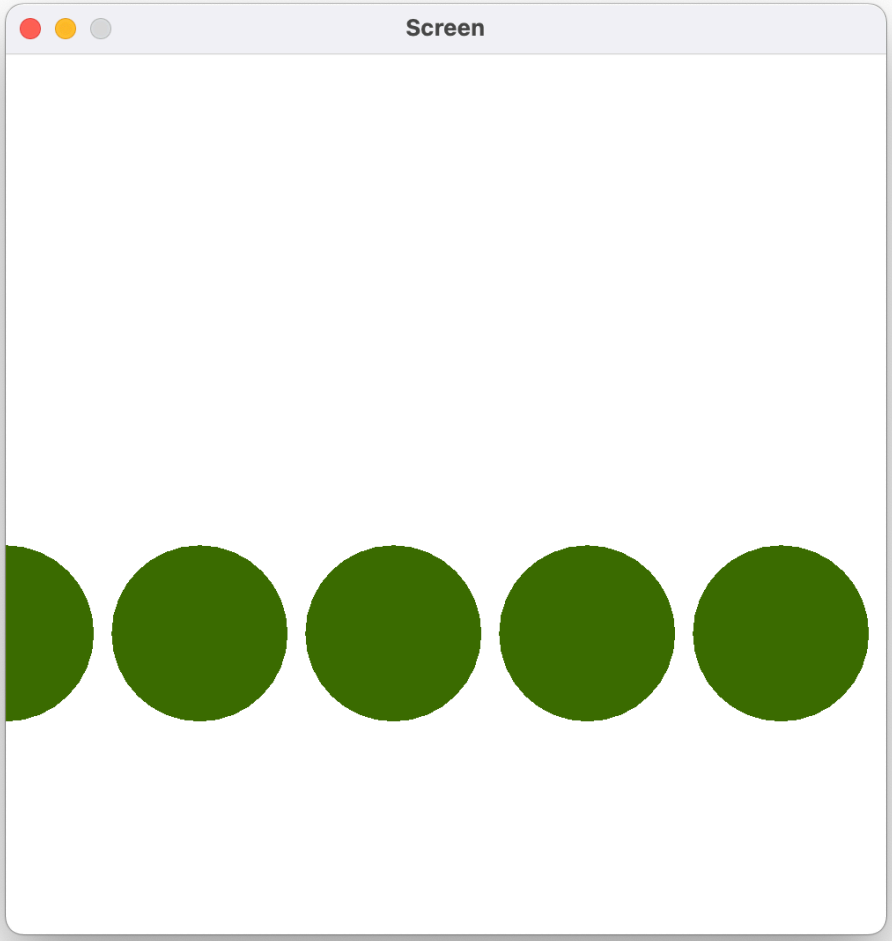
```
void circleLine() {
    int x = 0;
    int size = 50;
    int space = 10;
    while (x < screen.width) {
        fillCenteredCircle(x, 200, size);
        x = x+size+space;
    }
}
```

AN INPUT PARAMETER

```
void circleLine(int size) {
    int x = 0;
    int space = 10;
    while (x < screen.width) {
        fillCenteredCircle(x, 200, size);
        x = x+size+space;
    }
}
```

ANOTHER INPUT PARAMETER

```
void circleLine(int size, int y) {
    int x = 0;
    int space = 10;
    while (x < screen.width) {
        fillCenteredCircle(x, y, size);
        x = x+size+space;
    }
}
```


Screen

questions?

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/