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/

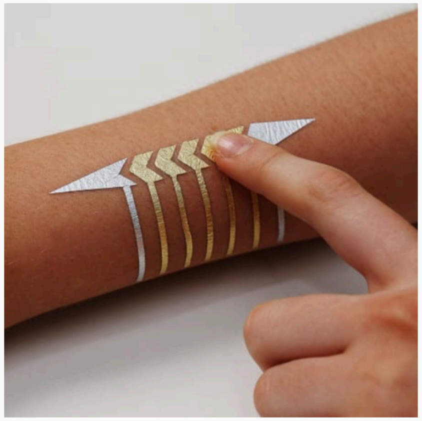
TALK TODAY: CINDY KAO

KNITDERMIS

WOVENPROBE

ESLUCENT

WOVENSKIN



DUOSKIN

SKINWIRE

HEADS UP

QUIZ 2 ON FRIDAY

CS TUTORING
FARRIS ENGINEERING CENTER, 2065

OPEN IntelliJ

CHANGE NAME OF PROJECT

WEEK5 - WEEK6

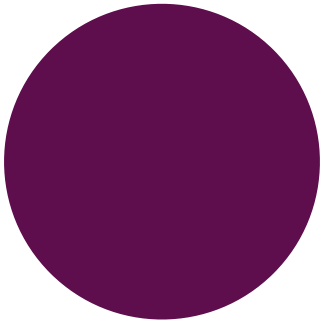
TODAY: STATIC METHODS

NON-STATIC METHODS

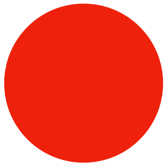
- Are associated with a specific object
- Manipulate instance variables
- ie:

```
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;
}
```

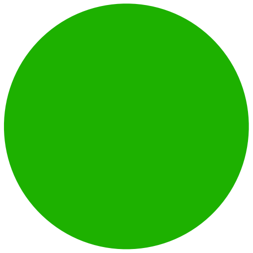
DIFFERENT BALL OBJECTS CREATED BY CONSTRUCTOR



```
color = purple
size = 150
xPosition = 200
yPosition = 200
```



```
color = red
size = 50
xPosition = 60
yPosition = 60
```



```
color = green
size = 90
xPosition = 307
yPosition = 124
```

NON-STATIC METHODS

- Are associated with a specific object
- That object is “this”, the current object
- “this” can be explicit or implicit
- ie: in Ball class:

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

=

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

NON-STATIC METHODS

- Are associated with a specific object
- That object is “this”, the current object
- “this” can be explicit or implicit
- ie: in MyPanel class:

```
MyPanel () {
    width = 500;
    height = 500;
    ball = new Ball(Color.BLACK, 60,300,212);
    ball.setSpeed(10,3);
    Dimension d = new Dimension(width,height);
    setPreferredSize(d);
    setVisible(true);
}
```

=

```
MyPanel () {
    this.width = 500;
    this.height = 500;
    this.ball = new Ball(Color.BLACK, 60,300,212);
    this.ball.setSpeed(10,3);
    Dimension d = new Dimension(width,height);
    this.setPreferredSize(d);
    this.setVisible(true);
}
```

questions?

WHAT DOES STATIC MEAN?

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    panel.animate(60);
}
```

WHAT DOES STATIC MEAN?

- Something that is shared between all objects in a class
- Can be a method or a variable

- Something that doesn’t fit the object oriented paradigm
- Something that doesn’t modify or refer to an object

EXAMPLE: MAIN METHOD

STATIC MAIN METHOD

- Not associated with a particular object
- The keyword “static” indicates that main() is not associated with any specific object
- There is no “this” in main

STATIC MAIN METHOD

- Because main is not associated with any particular object,
- Always have to be explicit about what object you are referring to.

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    panel.animate(60);
}
```

object created

STATIC MAIN METHOD

- Because main is not associated with any particular object,
- Always have to be explicit about what object you are referring to.

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    panel.animate(60);
}
```

specific object reference
when non-static method
is called

CODE THAT WILL CAUSE ERRORS

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    this.animate(60);
}
```

there is no this in main()

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    animate(60);
}
```

can’t directly access non-static methods
need to refer to a specific object

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    width = 400;
}
```

can’t directly access instance variables
need to refer to specific object

AN ILLUMINATING EXAMPLE

MAIN CAN HAVE MORE THAN ONE PANEL & FRAME

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyPanel panel2 = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    MyFrame frame2 = new MyFrame(panel2);
    panel.animate(60);
}
```

need to refer to a specific object
to know which panel to animate

questions?

ANOTHER ILLUMINATING EXAMPLE

ADD A STATIC METHOD TO MyPanel

```
static double square(double x) {
    double result = x*x;
    return result;
}
```

cannot refer to any instance variables inside the method

CALL THE STAIC METHOD IN MAIN

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    panel.animate(60);
    square(5);
}
```

don’t need to refer to an object

DIFFERENT WAYS TO CALL A STATIC METHOD IN MAIN

```java
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    panel.animate(60);
    square(5);
    MyPanel.square(6);
    panel.square(8);
}
```

no class or object reference

DIFFERENT WAYS TO CALL A STATIC METHOD

```
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    panel.animate(60);
    square(5);
    MyPanel.square(6);
    panel.square(8);
}
```

refer to Class instead of object

DIFFERENT WAYS TO CALL A STATIC METHOD

```java
public static void main(String[] args) {
    MyPanel panel = new MyPanel();
    MyFrame frame = new MyFrame(panel);
    panel.animate(60);
    square(5);
    MyPanel.square(6);
    panel.square(8);
}
```

refer to a specific object
will compile, but
bad coding practice

questions?

WHAT DOES STATIC MEAN/ WHEN TO USE “static”

- Something that is shared between all objects in a class
- Can be a method or a variable

- Something that doesn’t fit the object oriented paradigm
- Something that doesn’t modify or refer to an object

DELETE THE STATIC METHOD square
IF YOU ADDED ONE

WE’LL KEEP RETURNING TO THIS TOPIC

BACK TO BALLS…

ADD SOME RANDOMNESS

I WANT TO CREATE A BALL WITH
A RANDOM:
COLOR, SIZE, AND POSITION

EDIT THE FIRST BALL() CONSTRUCTOR

THE FIRST CONSTRUCTOR IN Ball

```
Ball () {
    color = Color.PINK;
    size = 50;
    xPosition = 200;
    yPosition = 200;
    xSpeed = 1;
    ySpeed = 1;
}
```

RANDOMIZING SIZE

```
Ball () {
    color = Color.PINK;
    size = 50;
    xPosition = 200;
    yPosition = 200;
    xSpeed = 1;
    ySpeed = 1;
}
```

what’s a good range for the size variable?

smallest size?

largest size?

RANDOMIZING SIZE

```
size = minSize + ??;
```

RANDOMIZING SIZE

```
size = minSize + (int)(Math.random() * ??);
```

max?

RANDOMIZING SIZE

```
size = minSize + (int)(Math.random() * (maxSize-minSize+1));
```

why +1?

behavior of Math.random

RANDOMIZING SIZE

```
Ball () {
    int minSize = 10;
    int maxSize = 100;
    size = minSize+(int)(Math.random()*(maxSize-minSize+1));
    color = Color.PINK;
    xPosition = 200;
    yPosition = 200;
    xSpeed = 1;
    ySpeed = 1;
}
```

RUN YOUR PROGRAM WITH THIS NEW CONSTRUCTOR

MyPanel

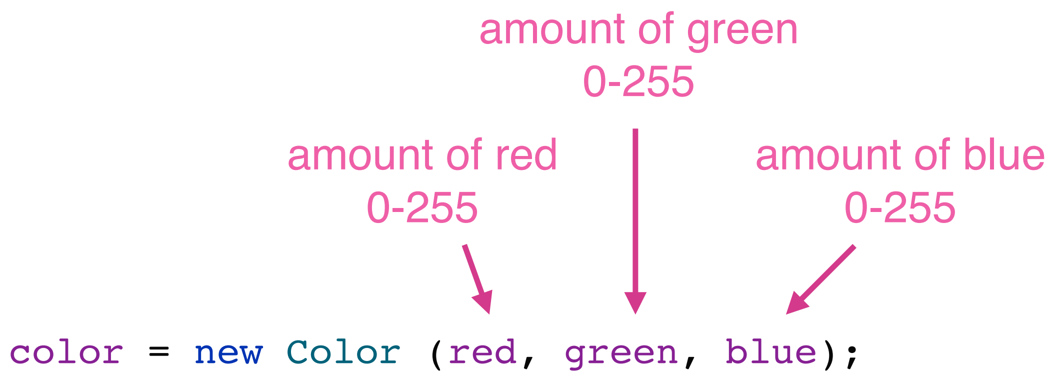
```
MyPanel () {
    width = 500;
    height = 500;
    ball = new Ball();
    ball.setSpeed(10,3);
    Dimension d = new Dimension(width,height);
    setPreferredSize(d);
    setVisible(true);
}
```

questions?

NOTE: STATIC METHOD

```
Ball () {
    int minSize = 10;
    int maxSize = 100;
    size = minSize+(int)(Math.random()*(maxSize-minSize+1));
    color = Color.PINK;
    xPosition = 200;
    yPosition = 200;
    xSpeed = 1;
    ySpeed = 1;
}
```

RANDOMIZING COLOR



amount of green
0-255

amount of red
0-255

amount of blue
0-255

```
color = new Color (red, green, blue);
```

RANDOMIZING COLOR

```
int red = ???;
int green = ???;
int blue = ???;
color = new Color (red, green, blue);
```

RANDOMIZING COLOR

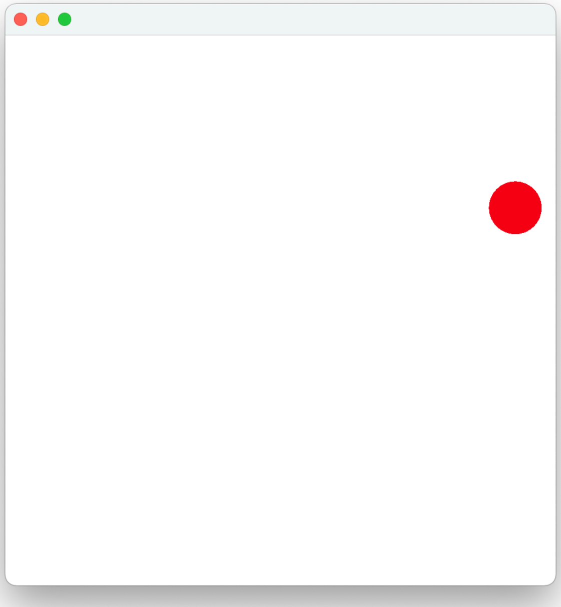
```
int red = (int)(Math.random()*256);
int green = (int)(Math.random()*256);
int blue = (int)(Math.random()*256);
color = new Color (red, green, blue);
```

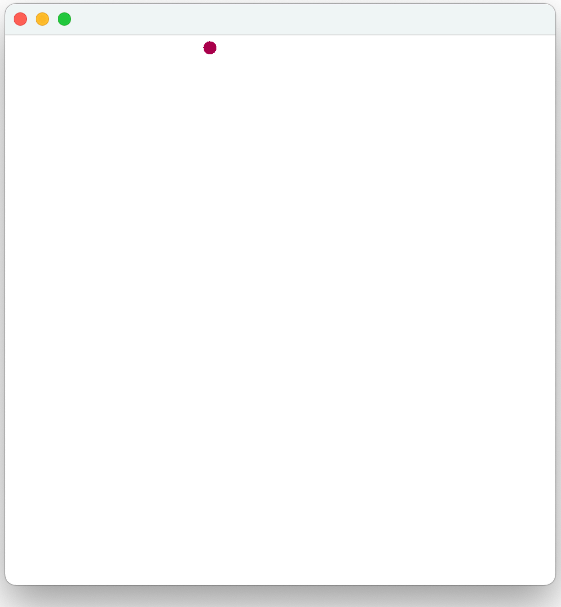
why 256 instead of 255?

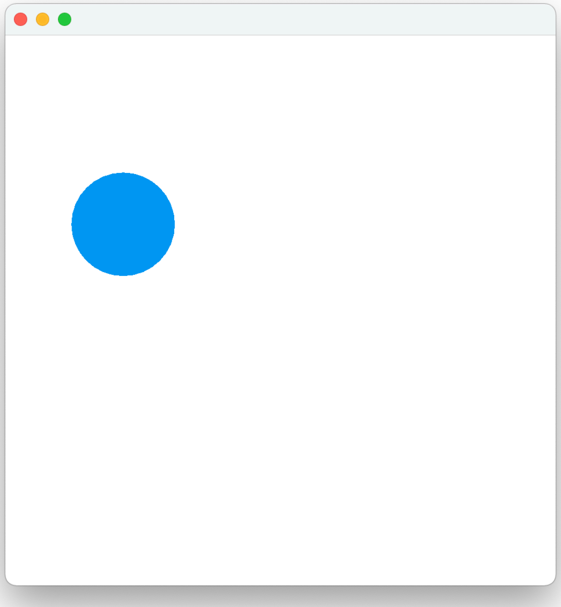
RANDOMIZING COLOR

```
Ball () {
    int minSize = 10;
    int maxSize = 100;
    size = minSize+(int)(Math.random()*(maxSize-minSize+1));
    int red = (int)(Math.random()*256);
    int green = (int)(Math.random()*256);
    int blue = (int)(Math.random()*256);
    color = new Color (red, green, blue);
    xPosition = 200;
    yPosition = 200;
    xSpeed = 1;
    ySpeed = 1;
}
```

RUN YOUR PROGRAM







questions?

RANDOMIZING POSITION

```
Ball () {
    int minSize = 10;
    int maxSize = 100;
    size = minSize+(int)(Math.random()*(maxSize-minSize+1));
    int red = (int)(Math.random()*256);
    int green = (int)(Math.random()*256);
    int blue = (int)(Math.random()*256);
    color = new Color (red, green, blue);
    xPosition = 200;
    yPosition = 200;
    xSpeed = 1;
    ySpeed = 1;
}
```

what do we need to know
to randomize position
if we want the ball to appear
in the window?

RANDOMIZING POSITION

```
Ball () {
    int minSize = 10;
    int maxSize = 100;
    size = minSize+(int)(Math.random()*(maxSize-minSize+1));
    int red = (int)(Math.random()*256);
    int green = (int)(Math.random()*256);
    int blue = (int)(Math.random()*256);
    color = new Color (red, green, blue);
    xPosition = 200;
    yPosition = 200;
    xSpeed = 1;
    ySpeed = 1;
}
```

window size!

RANDOMIZING POSITION: FIRST TRY

```
Ball (int windowWidth, int windowHeight) {
    int minSize = 10;
    int maxSize = 100;
    size = minSize+(int)(Math.random()*(maxSize-minSize));
    int red = (int)(Math.random()*256);
    int green = (int)(Math.random()*256);
    int blue = (int)(Math.random()*256);
    color = new Color (red, green, blue);
    xPosition = (int)(Math.random()*windowWidth+1);
    yPosition = (int)(Math.random()*windowHeight+1);
    xSpeed = 1;
    ySpeed = 1;
}
```

DELETE THE ENTIRE main() METHOD FROM THE BALL CLASS