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 FRIDAY

QUIZ 1 MOSTLY GRADED

TODAY: CLASSES AND OBJECTS

EXAMPLE: BALLS

what are some features of all balls?

- color
- size
- location
- speed

what are some things that balls do?

- move
- bounce
- spin

CLASS

defines features + behavior

CLASS

defines what a ball is, what it can do

NOT an actual ball

OBJECT

a single specific ball

- color = green
- size = 50 pixels
- location = (50, 100)
- speed = not moving

OBJECT

based on class template

- color = green
- size = 50 pixels
- location = (50, 100)
- speed = not moving

OBJECT

an “instance” of a class

- color = green
- size = 50 pixels
- location = (50, 100)
- speed = not moving

CLASSES AND OBJECTS

a way to combine
features (variables) &
behavior (functions/methods)
in code

CLASS: PERSON

what are some features of people?

- height
- hair color
- eye color
- weight
- alive?
- political affiliation
- name

CLASS: PERSON

what are some things that people do?

- eat
- lie
- run
- walk
- jump
- sleep
- vote ….

OBJECT?

- Dave
- Leah
- Brad Pitt
- Amy Adams

questions?

OPEN IntelliJ
CREATE A NEW CLASS CALLED BALL

CREATE A BALL CLASS

public keyword

class keyword

class name must be the same as file name

```
public class Ball {
}
```

BASIC CLASS STRUCTURE

```
public class Ball {
    int myVariable1;
    double myVariable2;

    Ball() {

    }

    void myMethod() {

    }
}
```

variable declarations
"instance" variables

"constructor" method
has same name as class
has no return type

other methods

what are some features of all balls?

- color
- size
- location
- speed

BASIC CLASS STRUCTURE

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

variable declarations
“instance” variables
properties of object

BASIC CLASS STRUCTURE

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

    Ball() {

    }
}
```

variable declarations
“instance” variables
properties of object

constructor method
creates an object

what are some things that balls do?

- move
- bounce
- spin

BASIC CLASS STRUCTURE

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

    Ball() {

    }

   public void move() {

    }
}
```

variable declarations
“instance” variables
properties of object

constructor method
creates an object

move() method
moves the ball

GOOD CODING PRACTICE

create a class skeleton

(variables + method definitions)

before writing all of the code

focus on high level structure first

NOW LETS FILL THINGS IN & EXAMINE MORE CLOSELY

CONSTRUCTOR: INITIALIZE VARIABLES

```
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;
    }

   public void move() {

    }
}
```

constructor method
creates an object
“initializes” all variables

CONSTRUCTOR: CREATES AN OBJECT

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

- a method
- different structure from any other method
- no modifiers (public, etc.)
- no return type
- exactly the same name as class, ie: “Ball” not “ball”
- creates an object, an “instance” of the class
- implicit return type = class

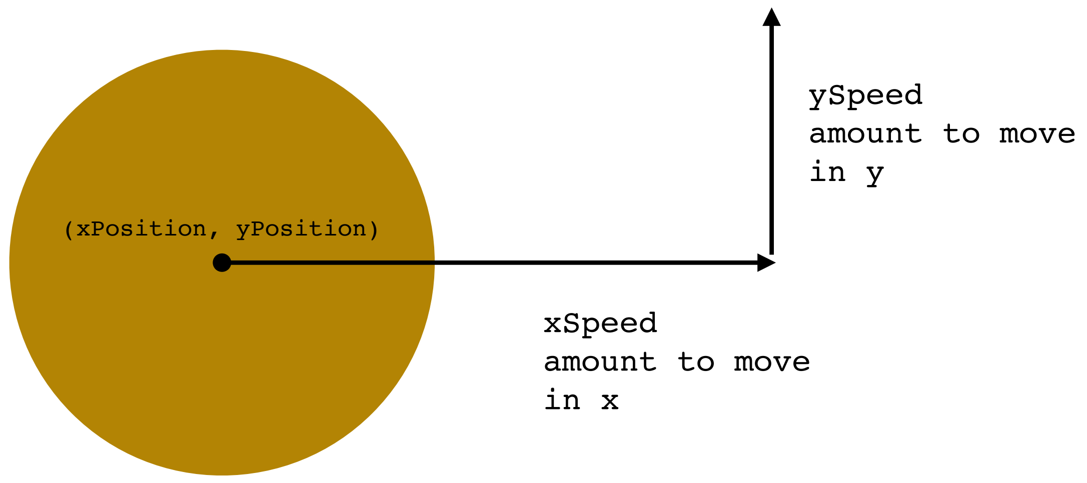
questions?

MOVE METHOD: CHANGES POSITION

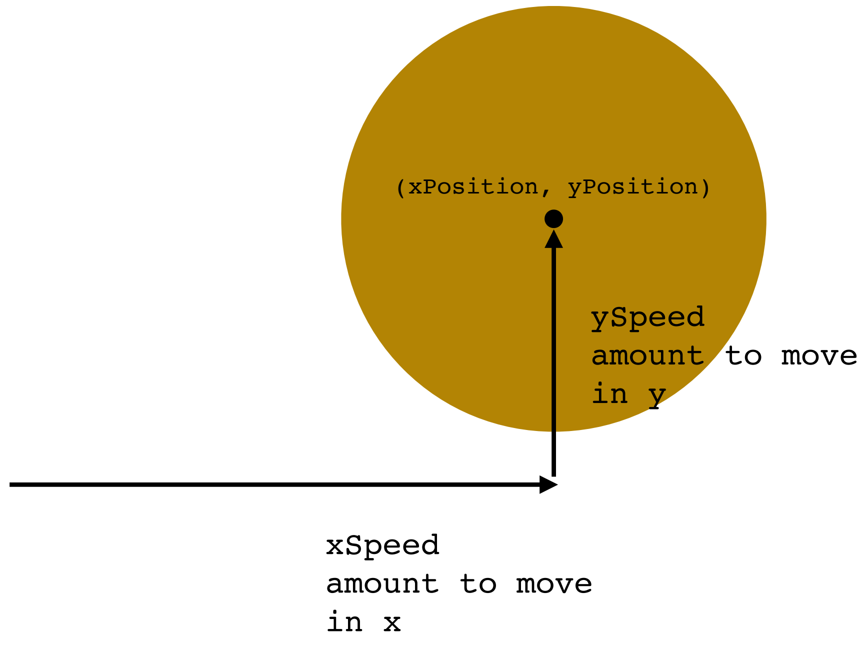
```
public class Ball {
    Color color;
    int  size;
    int xPosition;
    int yPosition;
    int xSpeed;
    int ySpeed;

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

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

move() method moves the ball, updating the position instance variables

BEFORE MOVE


(xPosition, yPosition)
ySpeed
amount to move
in y
xSpeed
amount to move
in x

AFTER MOVE


(xPosition, yPosition)
ySpeed
amount to move
in y
xSpeed
amount to move
in x

questions?

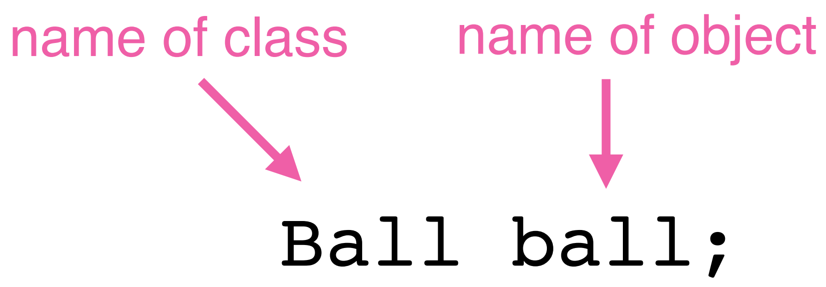
CREATE AN OBJECT

ADD A MAIN METHOD

```
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;
    }

    public static void main(String[] args) {
    }

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

main method
contains the code
that actually runs
the entry point

creating an object variable



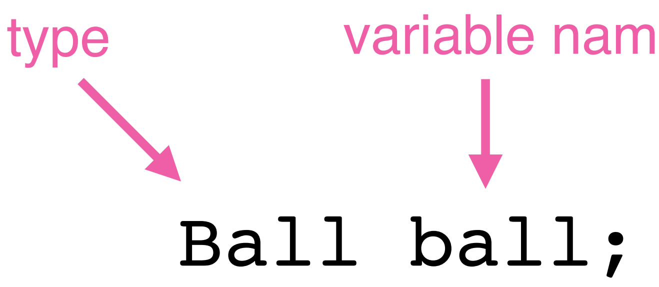
```
Ball ball;
```

in the computer's memory somewhere

ball

color	???
size	???
xPosition	???
yPosition	???
xSpeed	???
ySpeed	???

when you define a class you define a new TYPE



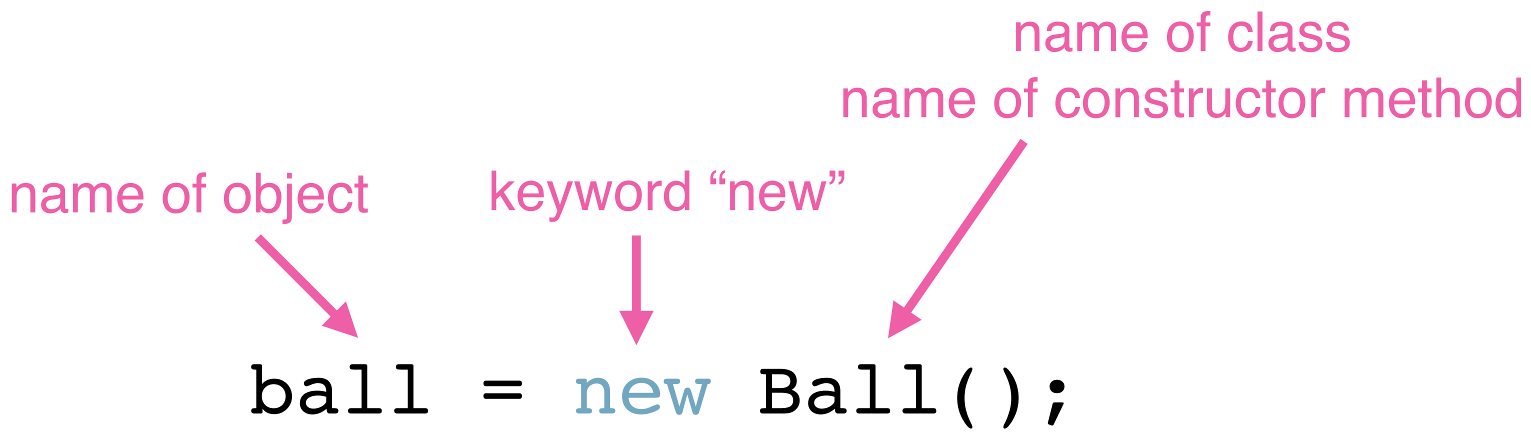
```
Ball ball;
```

in the computer’s memory somewhere

ball

color	???
size	???
xPosition	???
yPosition	???
xSpeed	???
ySpeed	???

creating a new object



```
ball = new Ball();
```

creating a new object calls the constructor method

```
ball = new Ball();
```

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

in the computer's memory somewhere

ball

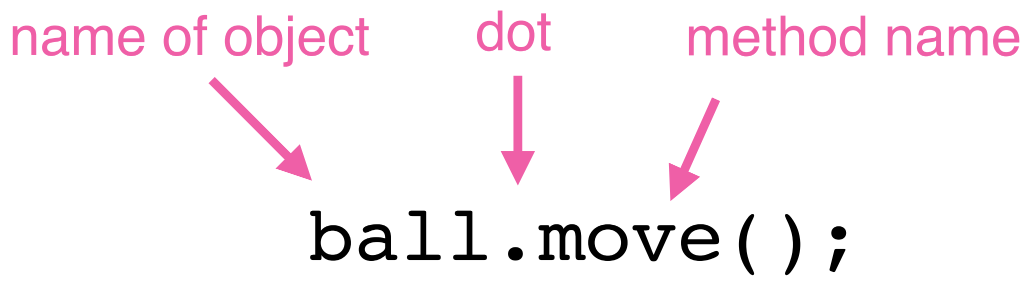
color	PINK
size	50
xPosition	100
yPosition	100
xSpeed	1
ySpeed	1

MAIN METHOD

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

MANIPULATING AN OBJECT

calling a method



```
ball.move();
```

```
variableName.method(method arguments);
```

MAIN METHOD

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

you can access instance variables using the same notation

name of object dot variable name

```
ball.xPosition
ball.ySpeed
ball.color
```

MAIN METHOD

```java
public static void main(String[] args) {
    Ball ball;
    ball = new Ball();
    System.out.println("xPosition: " +ball.xPosition);
    ball.move();
    System.out.println("xPosition after move: " +ball.xPosition);
}
```

```
xPosition: 100
xPosition after move: 101
```

questions?

WHAT IF I WANT
A BALL THAT’S A DIFFERENT SIZE?
A BALL THAT’S A DIFFERENT COLOR?

A SECOND CONSTRUCTOR

```
Ball(Color color, int size) {
    this.color = color;
    this.size = size;
    this.xPosition = 100;
    this.yPosition = 100;
    xSpeed = 0;
    ySpeed = 0;
}
```

A SECOND CONSTRUCTOR

```
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;
}
```

MAIN METHOD

```java
public static void main(String[] args) {
    Ball ball;
    ball = new Ball();
    ball.move();
    Ball ball2;
    ball2 = new Ball(Color.BLACK, 100, 300,300);
    System.out.println(“ball2 xPosition: " +ball2.xPosition);
    ball2.move();
    System.out.println(“ball2 xPosition after move: " +ball2.xPosition);
}
```

```
xPosition: 300
xPosition after move: 300
```

WHAT OTHER METHODS MIGHT BE USEFUL?

A SET SPEED METHOD

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

MAIN METHOD

```
public static void main(String[] args) {
    Ball ball;
    ball = new Ball();
    ball.move();
    Ball ball2;
    ball2 = new Ball(Color.BLACK, 100, 300,300);
    ball2.setSpeed(3,2);
    System.out.println("ball2 xPosition: " +ball2.xPosition);
    ball2.move();
    System.out.println("ball2 xPosition after move: " +ball2.xPosition);
}
```

```
xPosition: 300
xPosition after move: 303
```

A SET POSITION METHOD

```java
public void setPosition(int xPosition, int yPosition) {
    this.xPosition = xPosition;
    this.yPosition = yPosition;
}
```

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/