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/

MIDTERM WEDNESDAY

- 3 hours to complete exam
- 24 hour window
- 11am Wednesday - 11am Thursday
- Practice exam on Learn, under Exams
- Real exam will be a little shorter
- 10% of final grade

TODAY & WEDNESDAY: REVIEW

CLARIFICATION, CHEATING

- Test is open IntelliJ. Use it to write code, check answers, etc.
- Test is open internet. You can use the internet to help you solve challenges.
- This is an individual exam. Work should be entirely your own.
- Do not copy and paste code you find on the internet.
- No assistance from me, TAs, friends, or classmates for this exam
- Do not help or write code for anyone else for this exam
- I am grading the exam

EXAM TOPICS OVERVIEW

- Programming process: write, compile execute
- Variables
- Type
- Conditionals: if, else statements
- Boolean operations
- Methods
- CS coordinate system
- Generating random numbers
- Arrays, 1D and 2D
- Classes and Objects

NOT ON THE EXAM

- static
- MyFrame, MyPanel, and Screen details
- keyboard interaction using KeyListener
- getting input using scanner
- import class names and details

questions?

GRADING POLICY UPDATE:
WILL DROP LOWEST ASSIGNMENT/ QUIZ GRADE FROM AVERAGE

WILL GIVE YOU GRADE SO FAR AFTER MIDTERM

questions?

REVIEW: HARD STUFF FIRST

CLASSES AND OBJECTS

CLASSES AND OBJECTS

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

CLASS = TEMPLATE

OBJECT = ACTUAL THING

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

other methods

CONSTRUCTOR

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

- a method that creates an object, an “instance” of the class
- different structure from any other method
- no return type
- exactly the same name as class
- should initialize all “instance” variables

CONSTRUCTORS

```
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;
    colliding = false;
}
```

- a class can have more than one constructor
- each constructor has the same name
- each constructor must have different input parameters
- each constructor should initialize all instance variables

BASIC CLASS STRUCTURE

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

    Ball() {

    }

    Ball(Color color) {

    }

   public void move() {

    }
}
```

variable declarations
“instance” variables
properties of object

constructor methods
each one creates an object
each has different input parameters

other methods

A CLASS CAN HAVE A main() METHOD BUT DOESN'T HAVE TO

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

    Ball() {

    }

    Ball(Color color) {

    }

    public static void main(String[] args) {

    }

   public void move() {

    }
}
```

variable declarations
"instance" variables
properties of object

constructor methods
each one creates an object
each has different input parameters

optional main() method

other methods

QUIZ 2 QUESTION

Write a "Person" class with the following components:

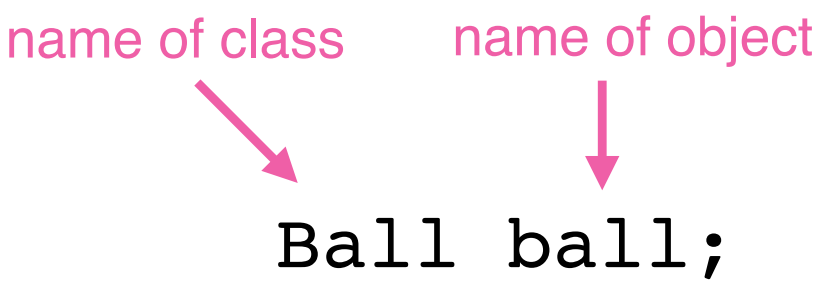
1) variables for:
- name
- age
- height (in inches)
- eyeColor

2) methods:
- a constructor that has no input parameters and initializes all variables to values that you choose
- a constructor that has name as an input parameter and uses this input to set the name. All other variables can be set to values that you choose.
- a method called setAge that takes age as an input parameter and uses this input to set the age

THERE WILL BE A QUESTION LIKE THIS ON THE EXAM

questions?

CREATING AN OBJECT: DECLARE A VARIABLE

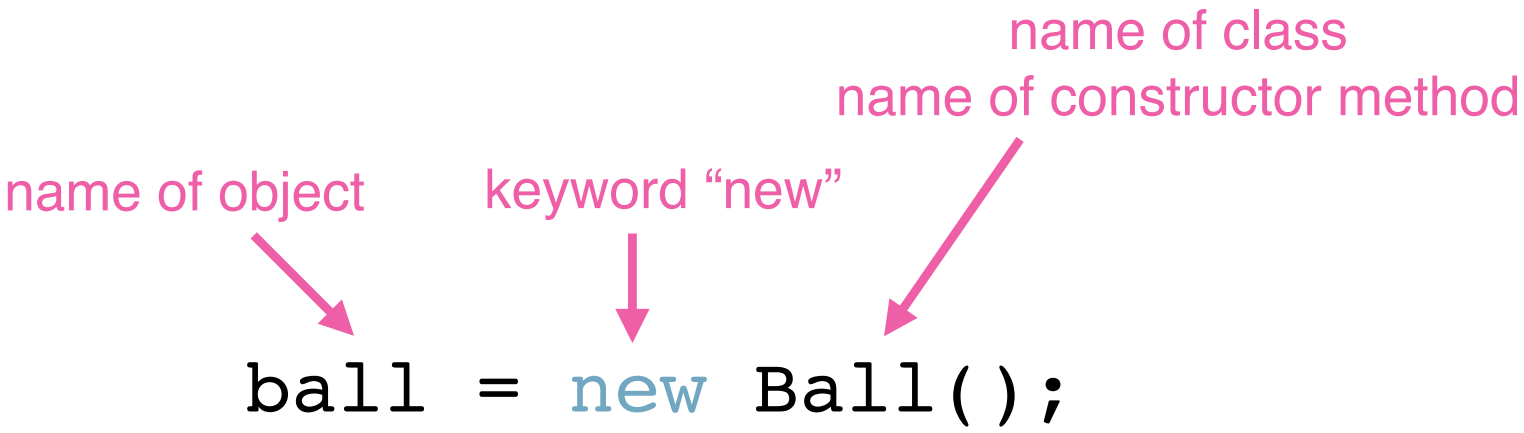


in the computer's memory somewhere

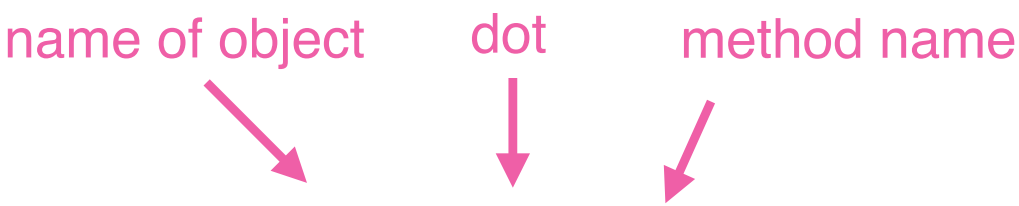
ball

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

CREATING THE OBJECT



CALLING A METHOD



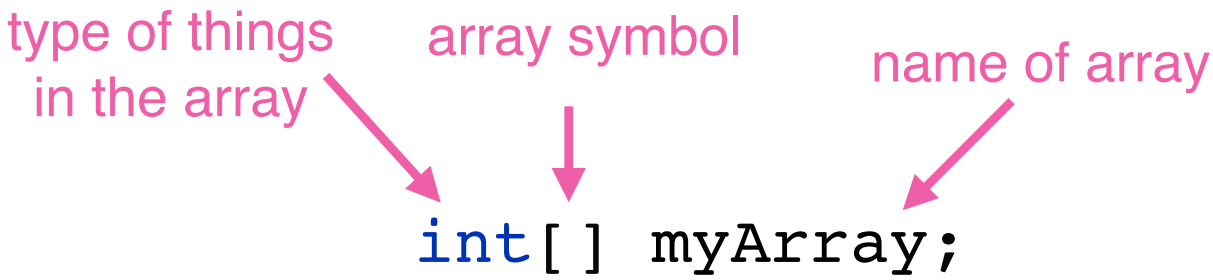

```
variableName.method(method arguments);
```

questions?

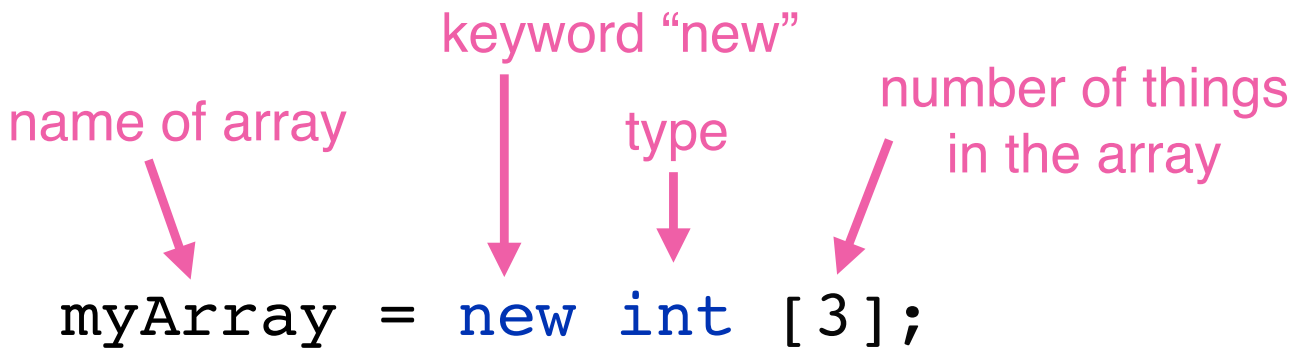
ARRAYS

ARRAYS ARE LISTS
OR COLLECTIONS OF THINGS
ALL OF THE SAME TYPE

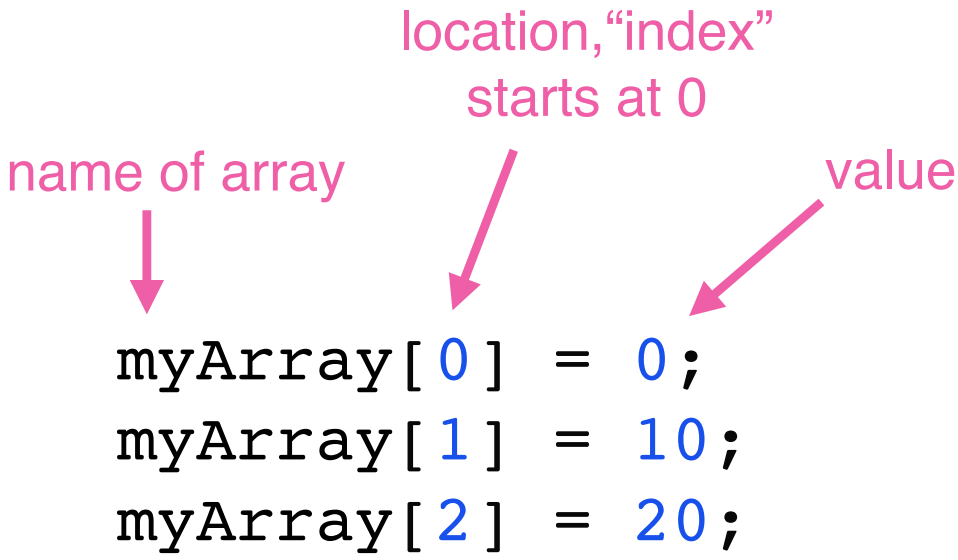
DECLARING AN ARRAY



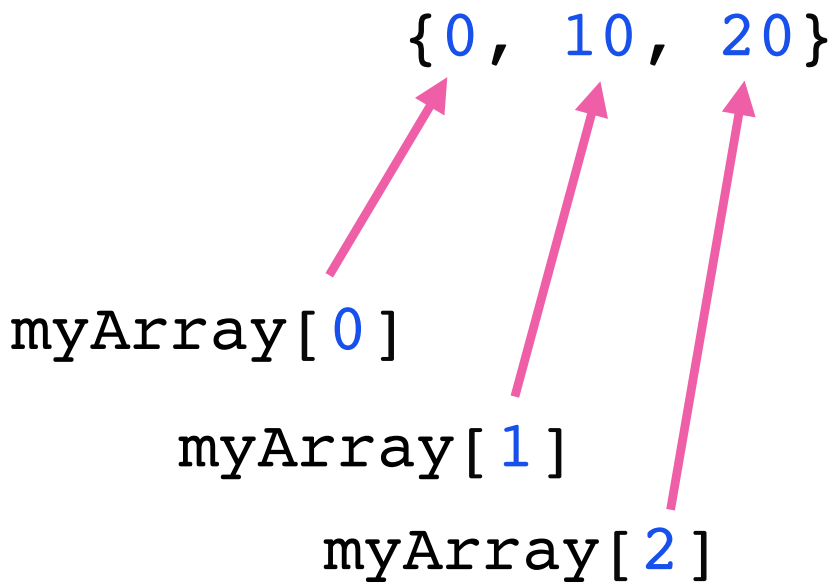
CREATING THE ARRAY



STORING VALUES IN AN ARRAY



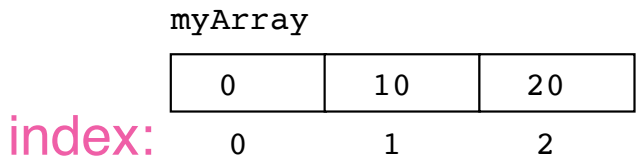
ACCESSING AN ARRAY ITEM



PUTTING IT ALL TOGETHER

```
int [] myArray;
myArray = new int [3];
myArray[0] = 0;
myArray[1] = 10;
myArray[2] = 20;
```

in the computer's memory somewhere



FOR LOOPS ❤ ARRAYS

condition: (i < length of array)

```
for (int i=0; i<3; i++) {
    myArray[i] = i*10;
    System.out.println(myArray[i]);
}
```

use i to reference array items
loop will access every item in array

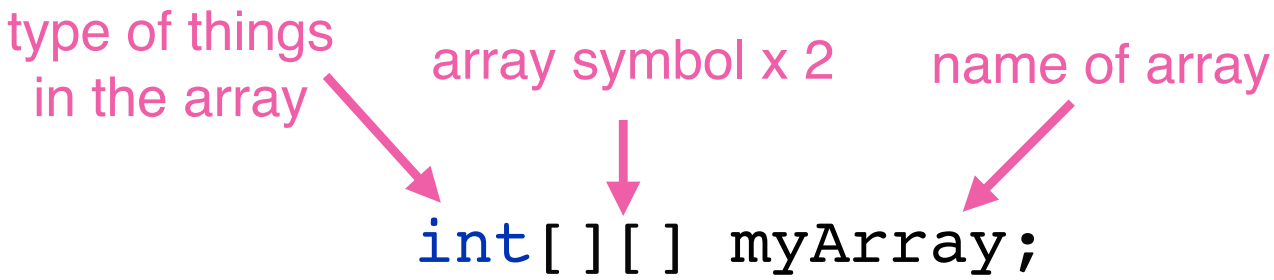
FOR LOOPS ♥ ARRAYS

condition: (i < length of array)

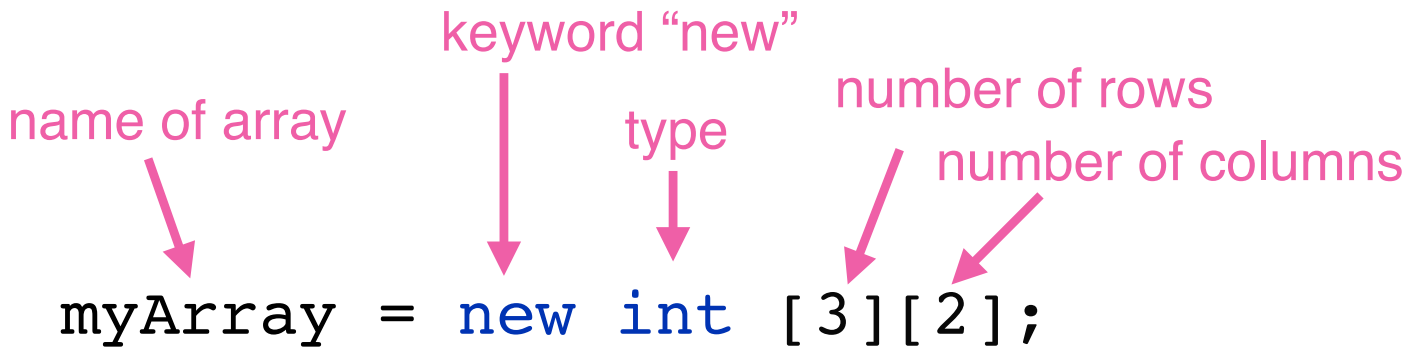
```
for (int i=0; i<myArray.length; i++) {
    myArray[i] = i*10;
    System.out.println(myArray[i]);
}
```

2D ARRAYS

DECLARING A 2D ARRAY

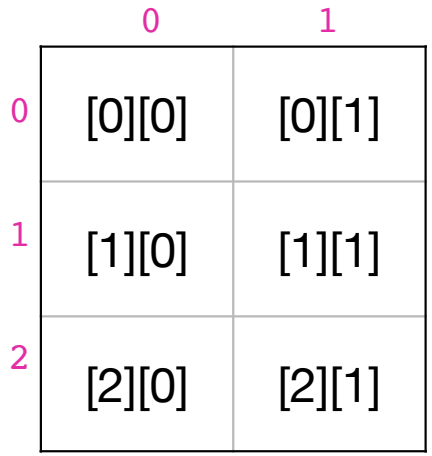


CREATING THE ARRAY

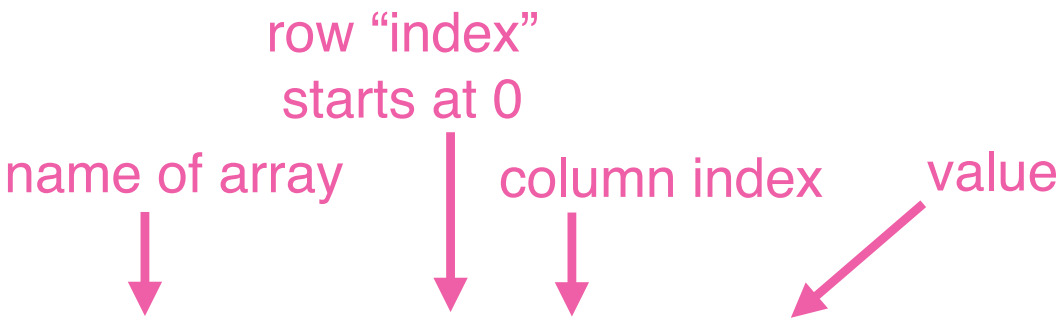


2D ARRAY INDICES

`myArray[row][column]`



STORING VALUES IN AN ARRAY




```
myArray[0][0] = 0;
myArray[0][1] = 10;
myArray[1][0] = 10;
myArray[1][1] = 20;
myArray[2][0] = 20;
myArray[2][1] = 30;
```

2D ARRAYS ❤ NESTED FOR LOOPS

```
int rows = 3;
int columns = 2;
int [][] myArray = new int [rows][columns];

for (int i=0; i<rows; i++) {

    for (int j=0; j<columns; j++) {
        myArray[i][j] = i*10 + j*10;
    }

}
```

this loop and variable i is for the rows

this loop and variable j is for the columns

GOOD PRACTICE WITH 2D ARRAYS: USE VARIABLES IN FOR LOOPS

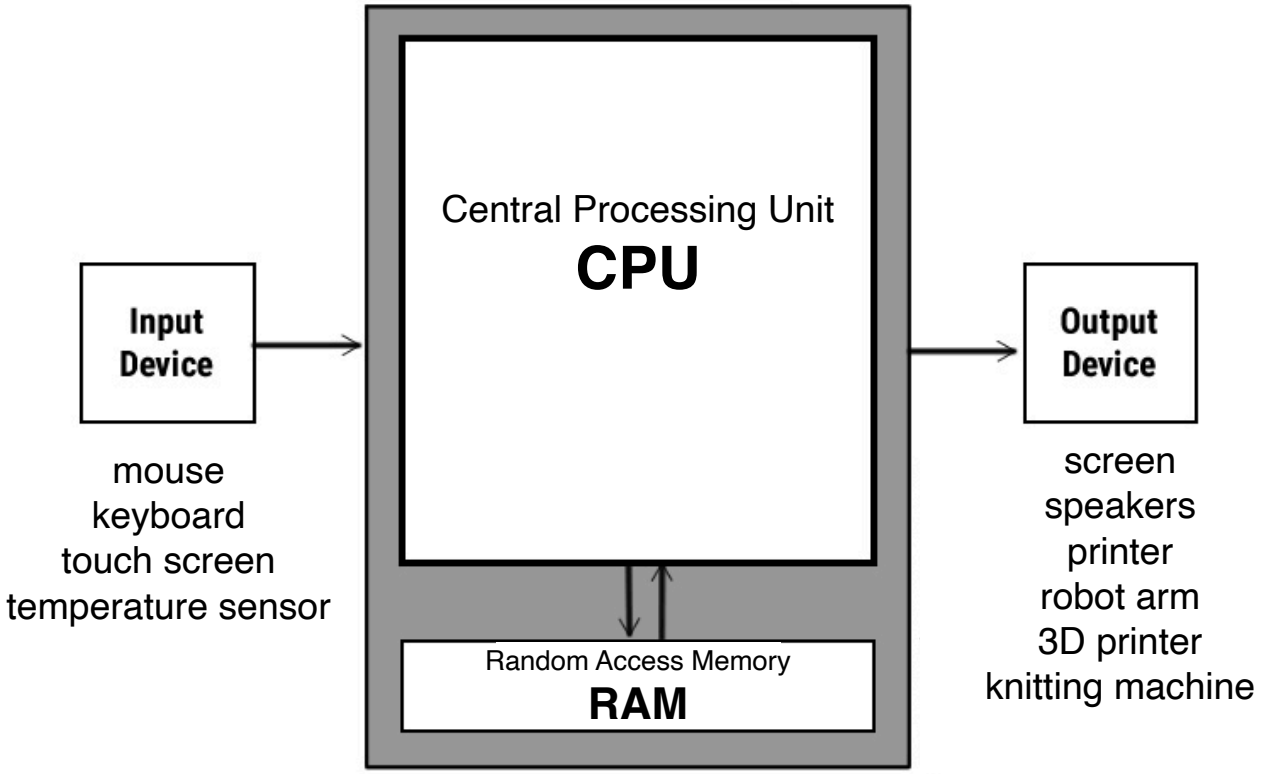
```
int rows = 3;
int columns = 2;
int [][] myArray = new int [rows][columns];

for (int i=0; i<rows; i++) {
    for (int j=0; j<columns; j++) {
        myArray[i][j] = i*10 + j*10;
        System.out.print(myArray[i][j] + "\t");
    }
    System.out.println();
}
```

questions?

REVIEW: STARTING FROM BEGINNING

ELEMENTS OF A COMPUTER



PROGRAMMING PROCESS

1. Write “source” code
2. Compile code
   - Compiler translates code written in high level language into “byte” code that a computer understands
   - Code with “syntax” errors will not compile
   - Syntax error = programming version of spelling and punctuation mistakes
3. Execute code
   - Computer runs or “executes” byte code
   - Turns written instructions into behavior!

VARIABLES

variable's type

```
int rectWidth = 200;
```

int = integer
a whole number

variable's name

```
int rectWidth = 200;
```

variable's value

```
int rectWidth = 200;
```

```
int rectWidth;
rectWidth = 200;
```

can also define a variable on one line
and assign a value to it later

TYPE

WHAT IS TYPE?

tells the computer how much memory
a variable takes up +
what it can do with the variable

BASIC NUMBER TYPES IN JAVA

TYPE	# BITS	minimum value	maximum value	example
byte	8	-128	127	53
int	32	-2,147,483,648	2,147,483,647	3079
float	32	$\sim -3.4 \times 10^{38}$ with 7 significant digits	$\sim 3.4 \times 10^{38}$ with 7 significant digits	4.589

MORE NUMBER TYPES IN JAVA

TYPE	# BITS	minimum value	maximum value	example
short	16	-32,768	32,767	134
long	64	~ -9.2×10^{18}	~ 9.2×10^{18}	30,790
double	64	~ -1.7×10^{308} with 15 significant digits	~ 1.7×10^{308} with 15 significant digits	10,789.998

OTHER “PRIMITIVE” TYPES IN JAVA

TYPE	# BITS	# possible values	examples
char	16	65,536	`'A'  'c' '?'`
boolean	1	2	`true false`

PRIMITIVE TYPE

cannot be broken down into a simpler type

MORE COMPLEX TYPES ARE COLLECTIONS OF THINGS

TYPE	collection of	example
String	chars	`String s = "hello";`
arrays	many items of a single type	`int[] n = {1,2,3};` `char[] c = {'a','b', 'c'};`
Color	numbers that define a color	`Color c = new Color(50,0,100);`

questions?