YEAH Hours

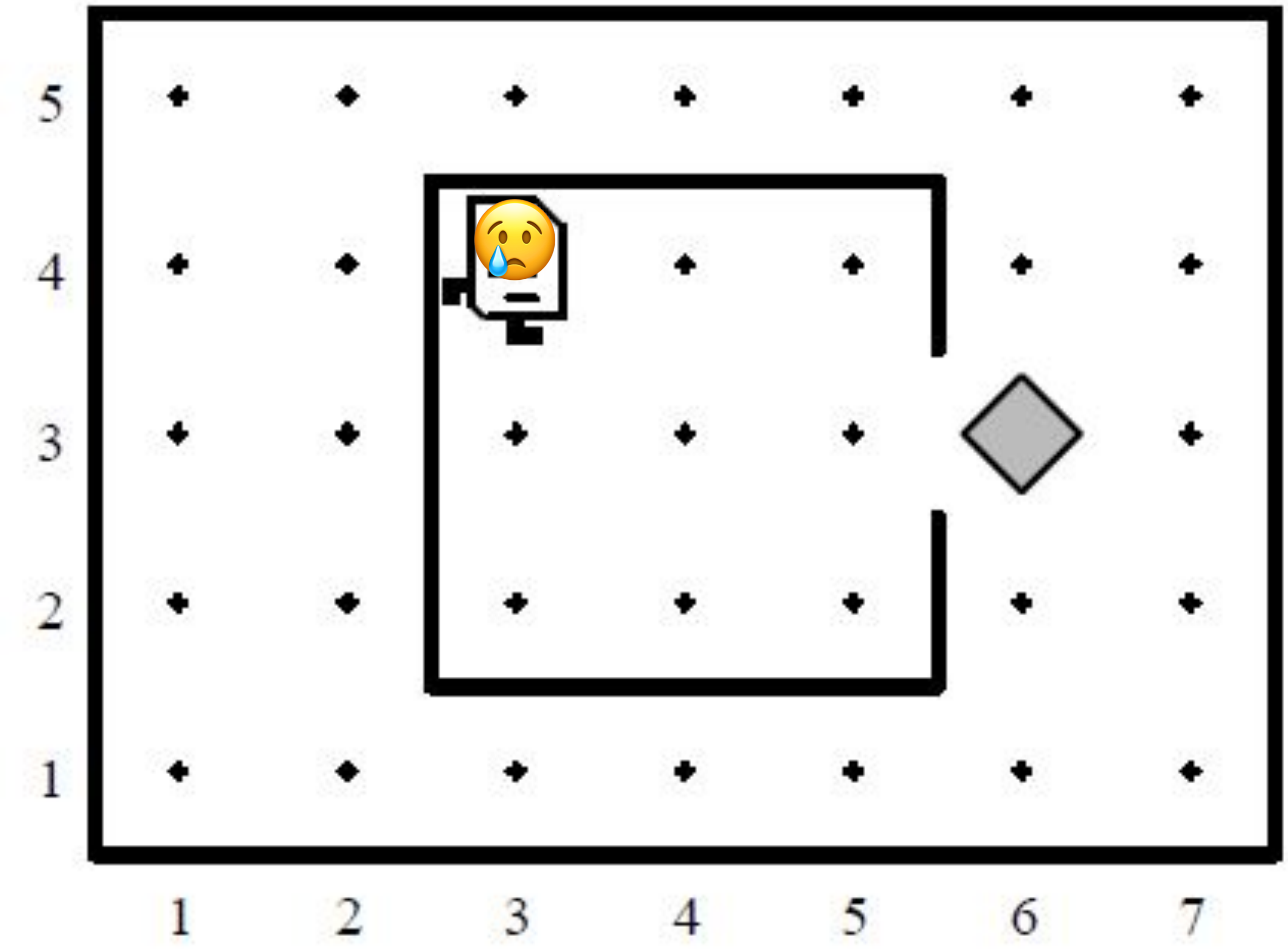
👍

January 23 2017, 7-8 PM

Jared Wolens

YEAH Hours?

- Held after each assignment is released
- Future dates to be scheduled soon
- Review + Assignment Tips
- Plan for today: lecture review, assignment tips, Q&A



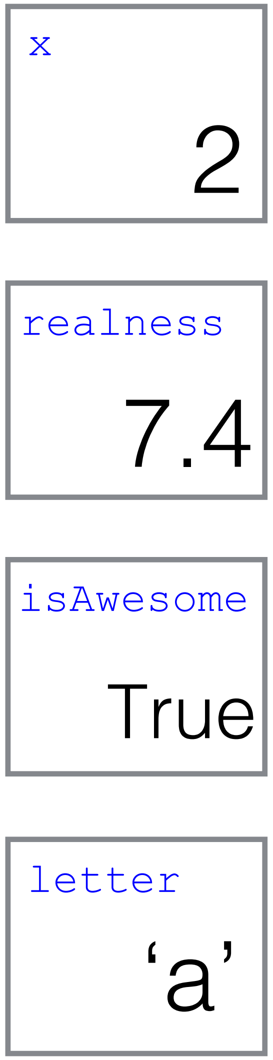
"Well, we're movin' on up (movin' on up)…"

- The Jeffersons

Variables

- **int**: Integers (counting)
- **double**: Real numbers (measuring)
- **boolean**: Logical true and false
- **char**: Letter, digit, and punctuation

```
int x = 2;
double realness = 7.4;
char letter = ‘a’;
boolean isAwesome = true;
```



x
2

realness
7.4

isAwesome
True

letter
‘a’

Variable Names

- ~~constant~~
- ~~void~~
- numDots
- sum
- ~~yourThing~~

Variable names should increase readability for ***everyone***

Boolean Review

- `!` “not”
  - If p is true then !p is false (and vice versa)
- `&&` “and”
  - Both sides must be true
- `||` “or”
  - Either side can be true

Constants

- Not all variables change; you may decide to have a variable representing a set value.
  - This should made a `constant`

```
UPPERCASE_WITH_UNDERSCORES
```

```
private static final double PI = 3.14159
private static final type name = value
```

Control Structures

🔁

for vs. while

```
for (init; test; step) {
   statements
}
```

- for loop used for **definite** iteration
- *Usual Case:* We know how many times we want to iterate

```
init
while (test) {
   statements
   step
}
```

- while loop used for **indefinite** iteration
- *Usual Case:* We don’t know how many times to iterate beforehand

Sentinel Values

```
while (true) {
  // …get a value from the user…
  if (condition) {
    break;
  }

  // …rest of body…
}
```

Use a sentinel's presence as a condition to ***break***

```
private static final int MAX_STEPS = 100
```

Example: Error Checking

```
int n;

while (true) {
   n = readInt(“Enter a positive integer: “);
   if (n > 0) {
      break;
   }

   println(“Invalid input.  Try again.”);
}

// n is now guaranteed to be positive!
```

Assignment 2!

💻

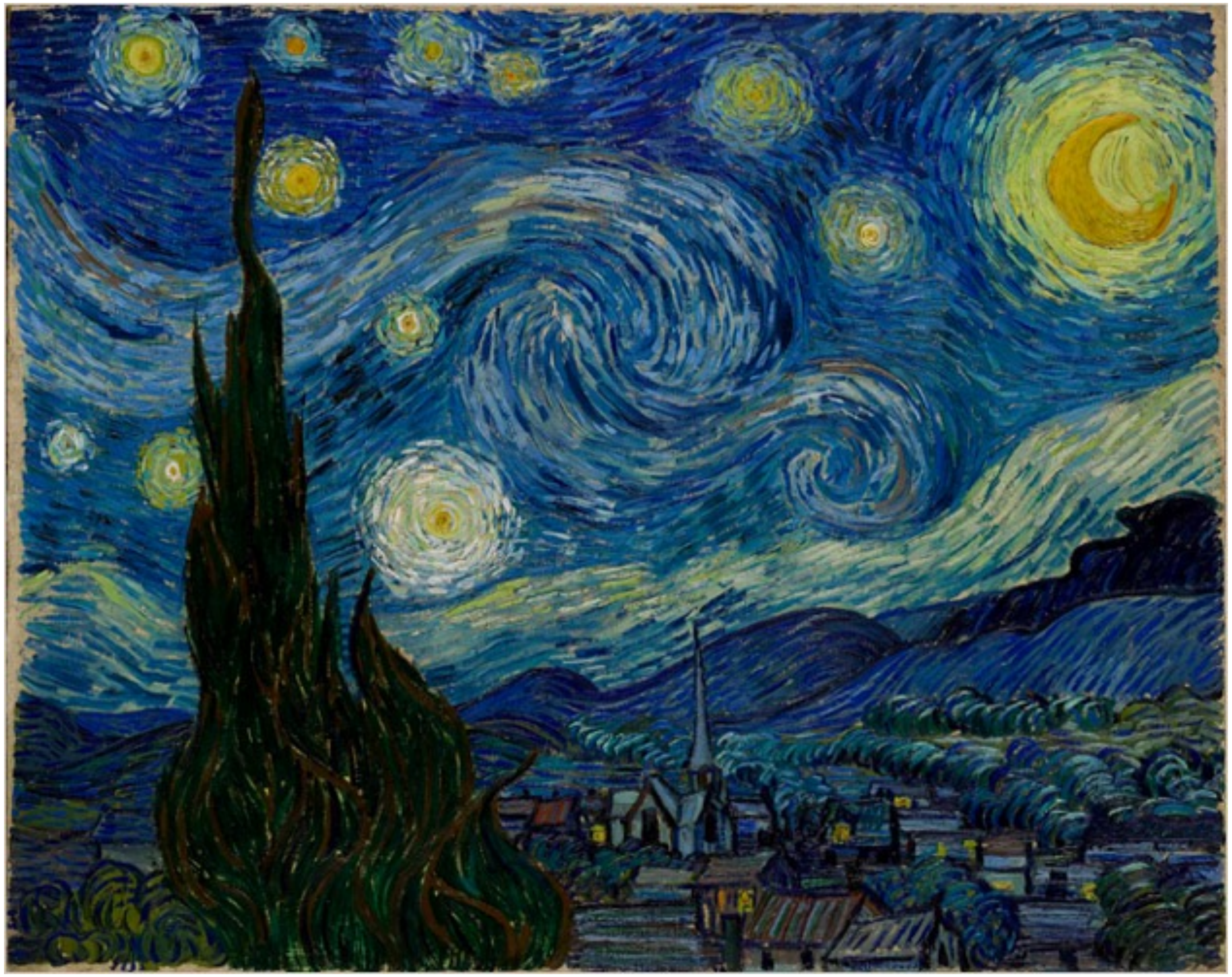
“READ THE SPECIFICATION CAREFULLY”

Assignment 2: Console & Graphics Programs

- Due Monday January 30 @ **10:30AM**
- *Warmup* graphics and console problem
- Then 3 are graphics, last 3 are console
- No particular order of difficulty
- Key for style: make sure you decompose! Might want to wait until Wednesday for some parts

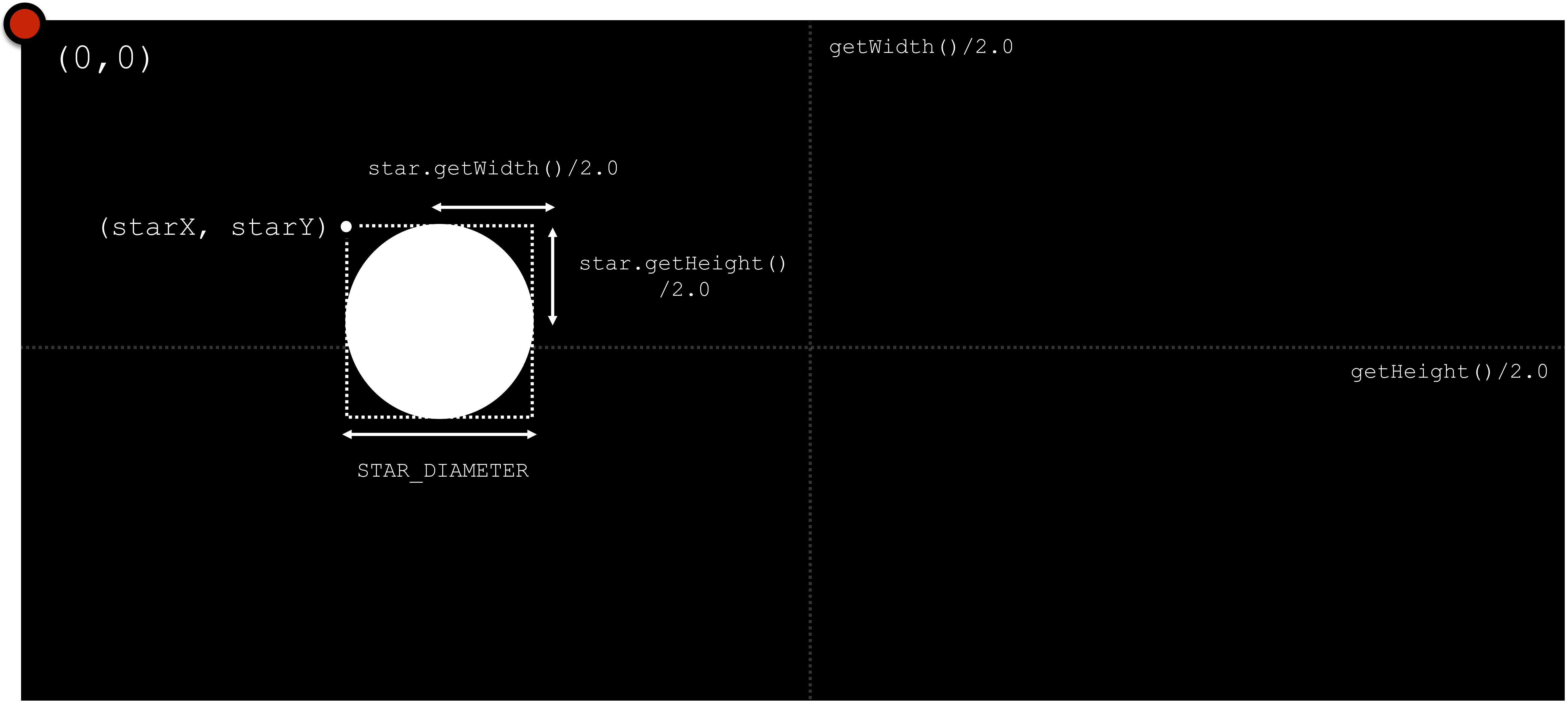
Graphics Warmup

🎨



“The Starry Night” (June 1889), by Vincent van Gogh

```
private void drawStar() {
    GOval star = new GOval(STAR_DIAMETER, STAR_DIAMETER);
    star.setFilled(true);
    star.setColor(Color.WHITE);
    double starX = 40;
    double starY = 40;
    add(star, starX, starY);
}
```



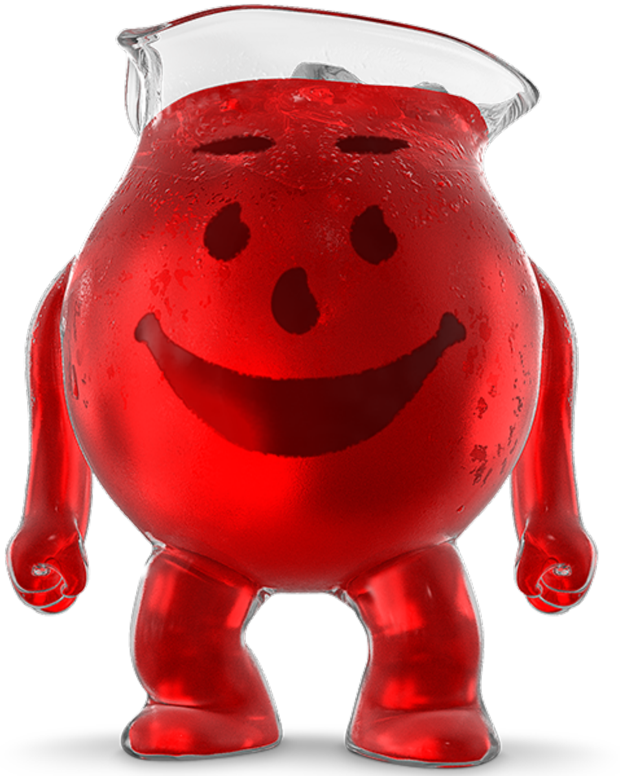
Console Warmup

🔢

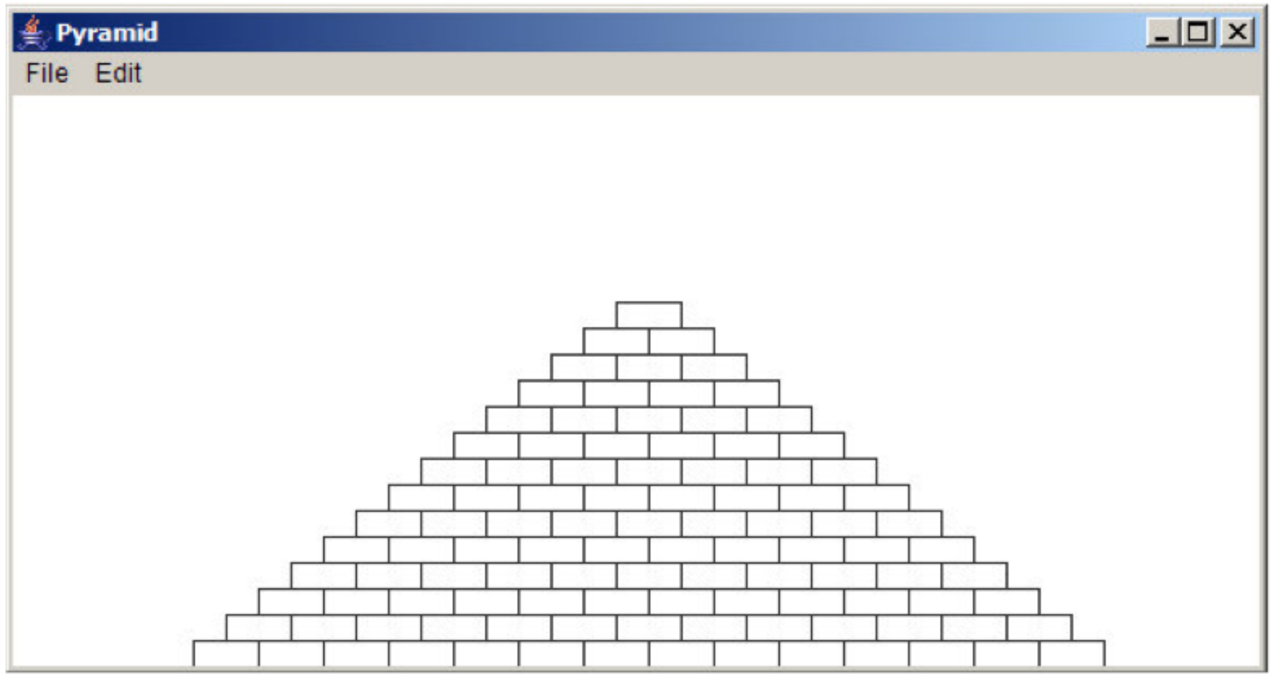
Print “YEAH!” 10 Times

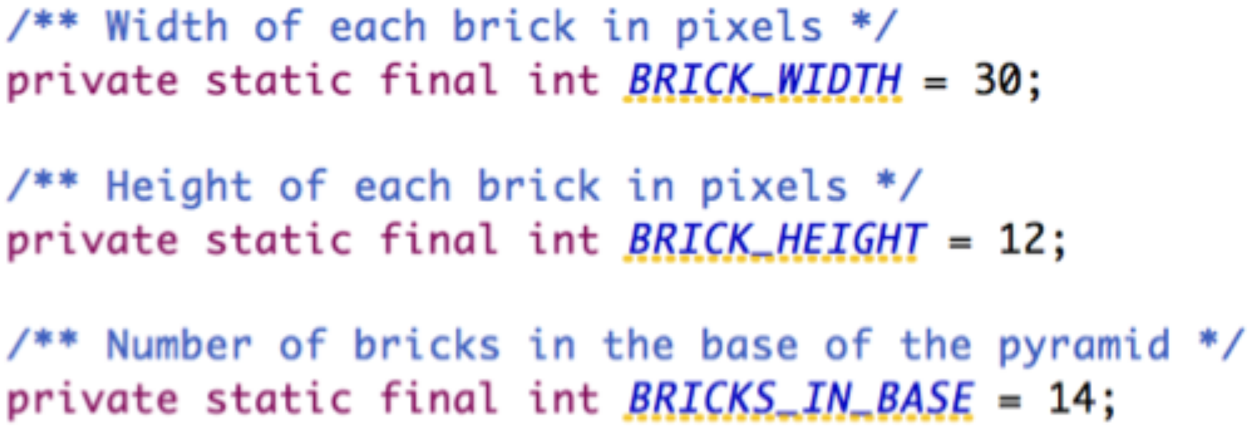
```
private void run() {
   for(int i = 0; i < 10; i++) {
     println(“YEAH!”);
   }
}
```

What’s the value of **i** when we stop?



1. Pyramid





```
/** Width of each brick in pixels */
private static final int BRICK_WIDTH = 30;

/** Height of each brick in pixels */
private static final int BRICK_HEIGHT = 12;

/** Number of bricks in the base of the pyramid */
private static final int BRICKS_IN_BASE = 14;
```

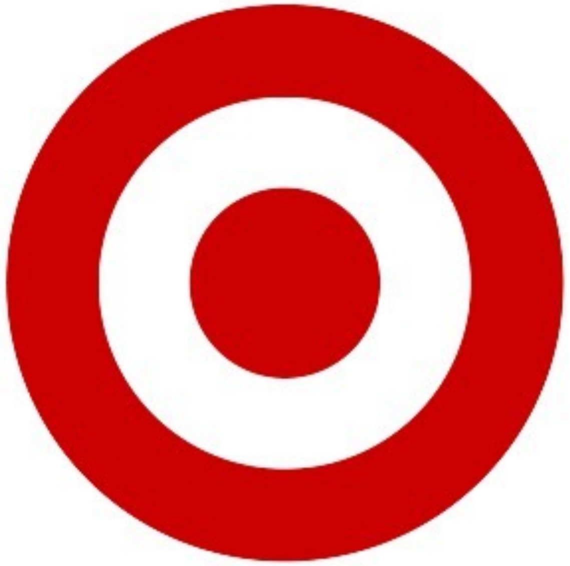
- Start by trying a *row*, then two rows

- Testing: Try changing the given **constants** often

- Extensions?

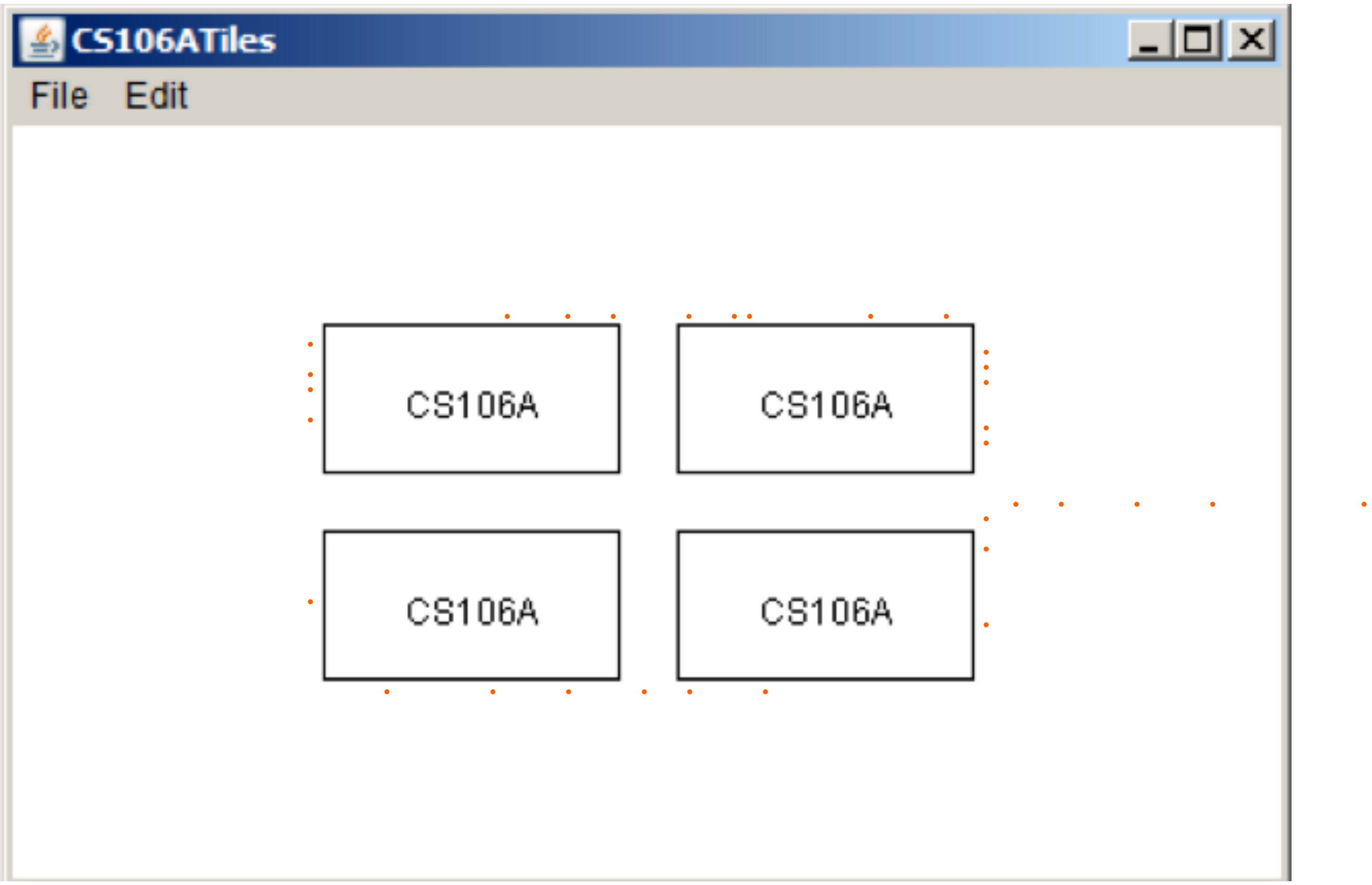
2. Target

*The outer circle should have a radius of one inch (72 pixels), the white circle has a radius of 0.65 inches, and the inner red circle has a radius of 0.3 inches… **centered**…*



- What is actually **changing** between each circle?
- **Decompose** the problem so you don't copy & paste code
- Circle border color
- Testing: Try changing the given circle sizes

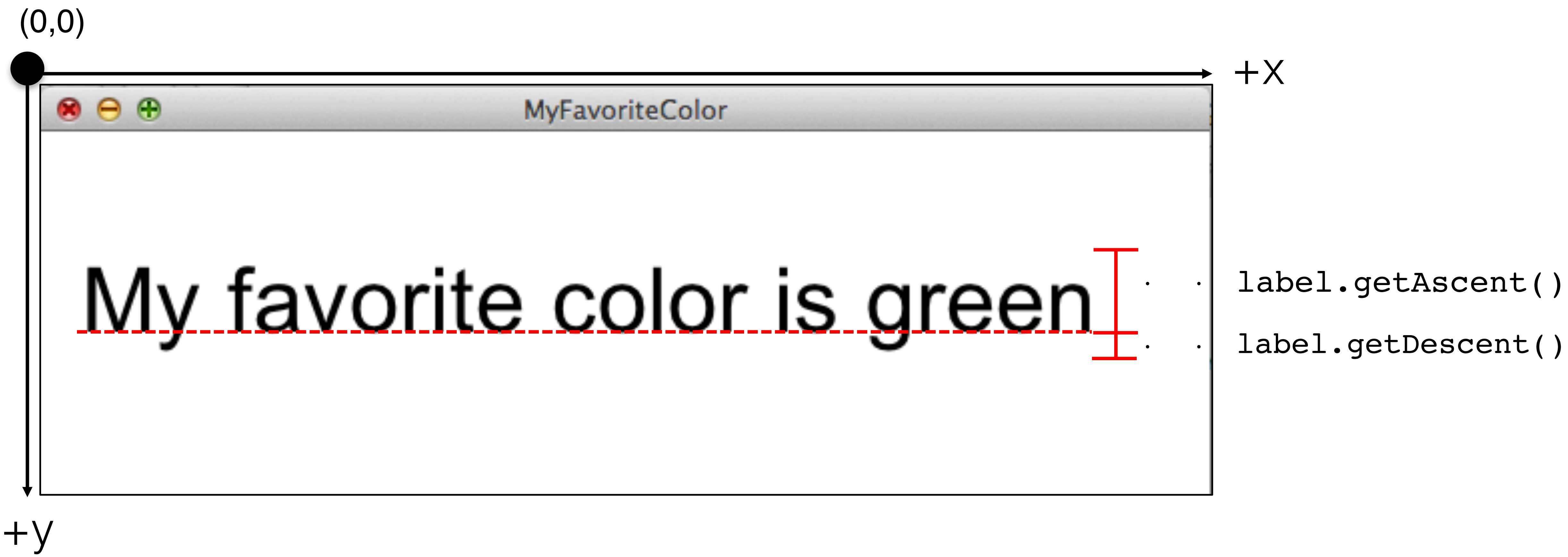
3. CS106A Tiles



imaginary
bounding box

- Think of it as **one big rectangle**
- `TILE_WIDTH, TILE_HEIGHT, TILE_SPACE`
- <u>Testing</u>: Try changing the given **constants**
- Centering GLabels (baseline of first character)

GLabels



```
GLabel label = new Glabel(“My favorite color is green”);
// use label.getAscent(), not label.getHeight()!
// (that way label is centered according to baseline)
double x = getWidth()/2.0 – label.getWidth()/2.0;
double y = getHeight()/2.0 + label.getAscent()/2.0;
// label size depends on text and font – can only center
// AFTER creating the label
add(label, x, y);
```

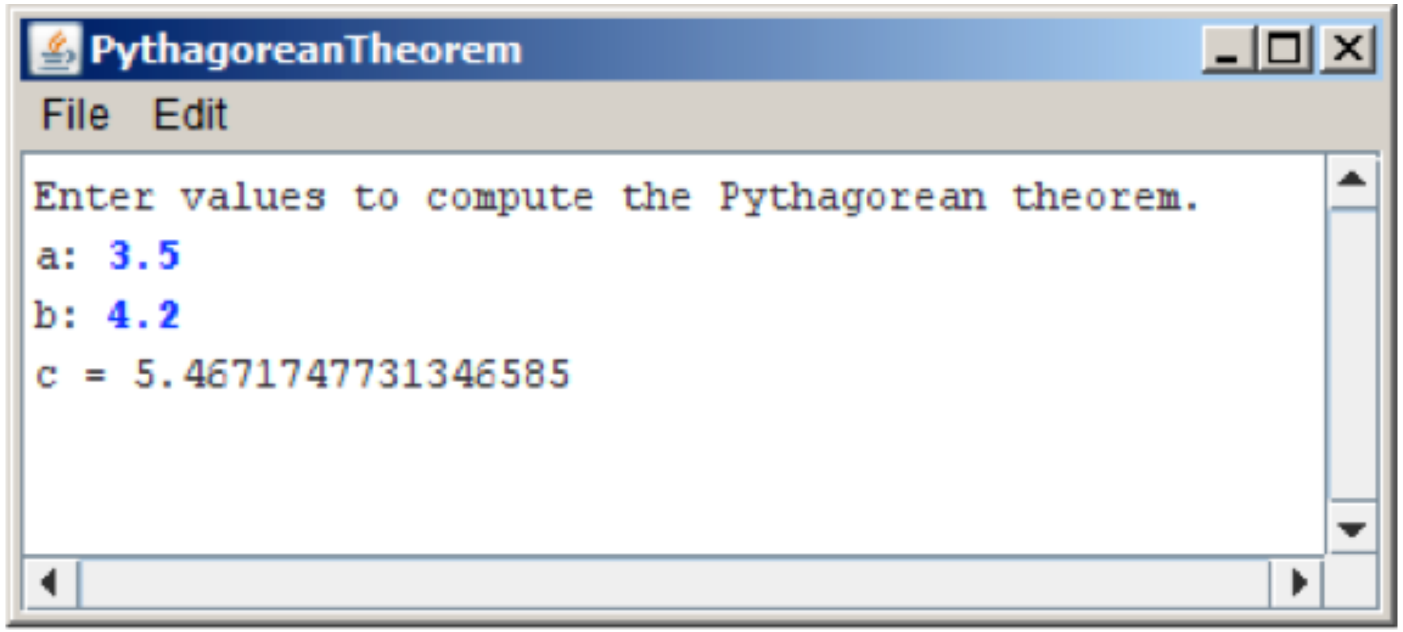
General Graphics Tips

- Draw pictures! Many graphics problems are just simple geometry in disguise

- Always use **`double`** when calculating coordinates

- getWidth() and getHeight() are your friends

4. Pythagorean Theorem



```
Enter values to compute the Pythagorean theorem.
a: 3.5
b: 4.2
c = 5.4671747731346585
```

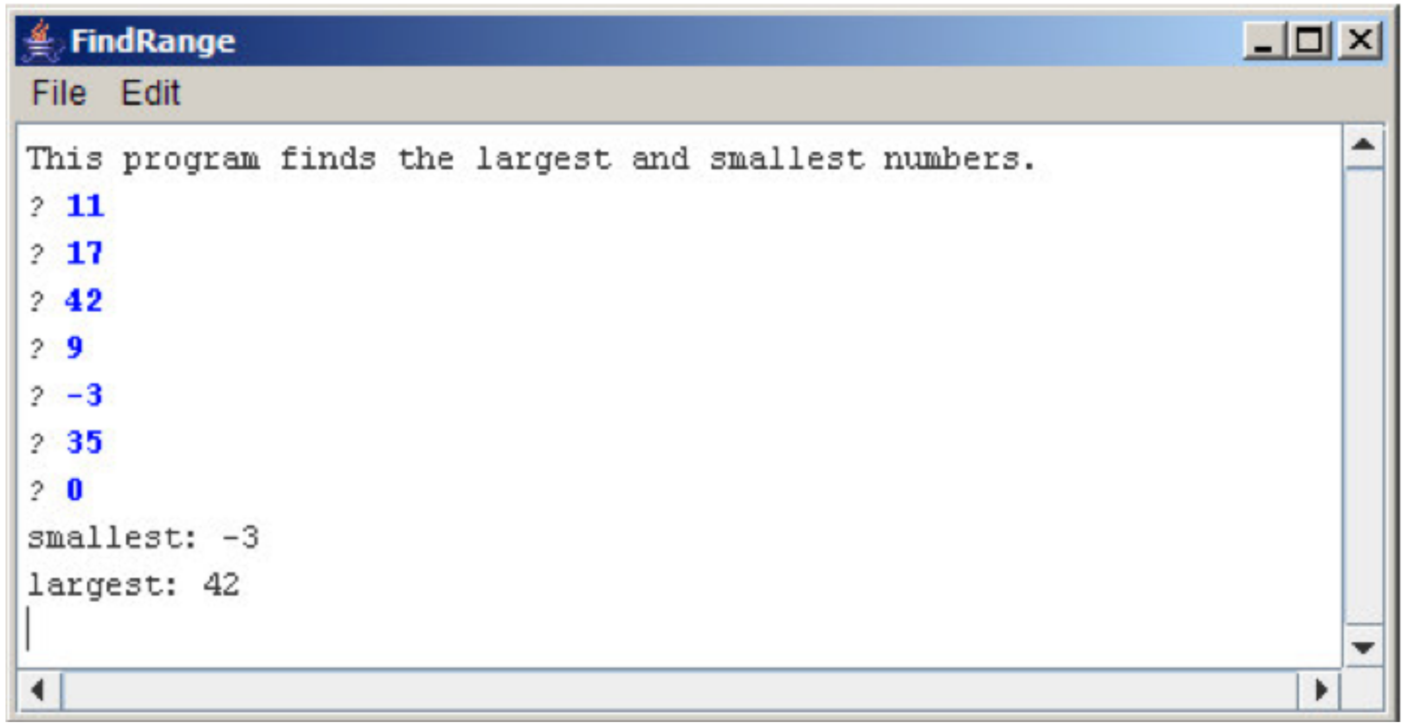
$$c = \sqrt{a^2 + b^2}$$

```
double y = Math.sqrt(x);
```

- Can assume inputs are positive
- Use **double**!
- Order of operators in Java: * (**m**ultiplication), / (**d**ivision), + (**a**ddition), - (**s**ubtraction)

“The Midas Touch”

5. Max/Min



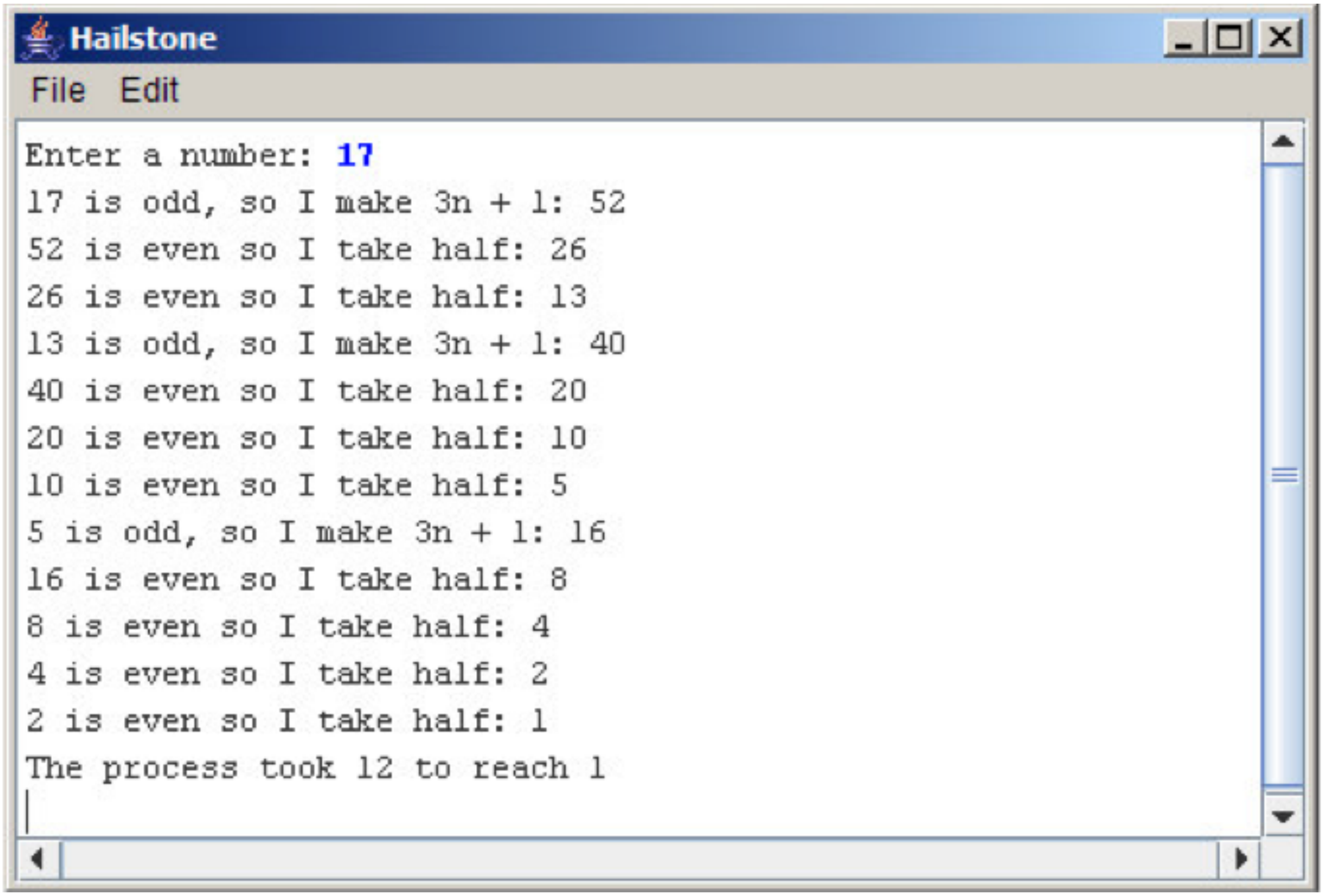
```
This program finds the largest and smallest numbers.
? 11
? 17
? 42
? 9
? -3
? 35
? 0
smallest: -3
largest: 42
```

If the user enters only 1 value before the sentinel, the program should report that score is the max and min.

If the user enters the **`sentinel`** on the very first input line, then no scores have been entered, and your program should tell the user that no values have been entered.

- Use variables (what type?) to determine the min and max
- Special cases!
- Use a constant for the **`sentinel`** (0)
- Testing: one number, negative numbers, no numbers

6. Hailstone



```
Enter a number: 17
17 is odd, so I make 3n + 1: 52
52 is even so I take half: 26
26 is even so I take half: 13
13 is odd, so I make 3n + 1: 40
40 is even so I take half: 20
20 is even so I take half: 10
10 is even so I take half: 5
5 is odd, so I make 3n + 1: 16
16 is even so I take half: 8
8 is even so I take half: 4
4 is even so I take half: 2
2 is even so I take half: 1
The process took 12 to reach 1
```

Pick some integer and call it n. *If n is even, divide it by two. If n is odd, multiply it by three and add one. Continue this process until n is equal to one*.

- Determining **odd** and **even**
- Testing: 1, even, odd

The Remainder Operator

- a % b is pronounced “a **mod** b.”
  - 15 % 3 = 0
  - 14 % 8 = 6
  - 21 % 2 = 1
  - 14 % 17 = 14

15/3 = 5 remainder 0

14/8 = 1 remainder 6

21/2 = 10 remainder 1

14/17= 0 remainder 14

Final Tips

- Follow the specifications carefully
- Comment!
- Go to the **LaIR** if you get stuck
- Incorporate **IG** feedback!

- Have fun!

Q&A

🙋‍♀️

Board Notes

```
int x = 3;
int y = 5;
int z = 4;
if (z > x && z < y) {
  // executes!
}
```

A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

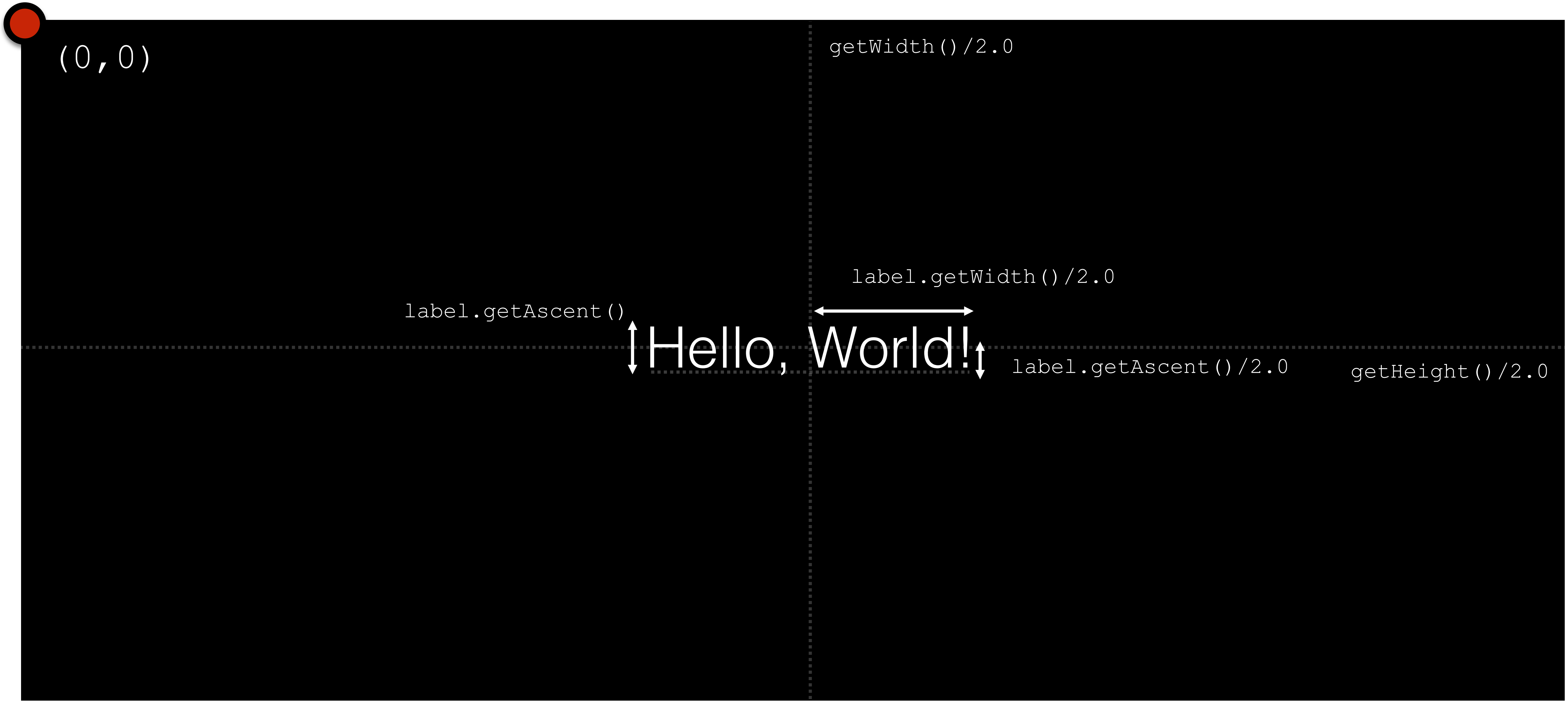
type of variable we `return`

```
private boolean functionName() {
 return true;
}
```

```
public void run() {
 while(functionName()) {
   // while(true) loop
 }
}
```

```
private static final int NUM_LIVES = 3
```

Board Notes



(0,0)
getWidth()/2.0
label.getWidth()/2.0
label.getAscent()
Hello, World!
label.getAscent()/2.0
getHeight()/2.0