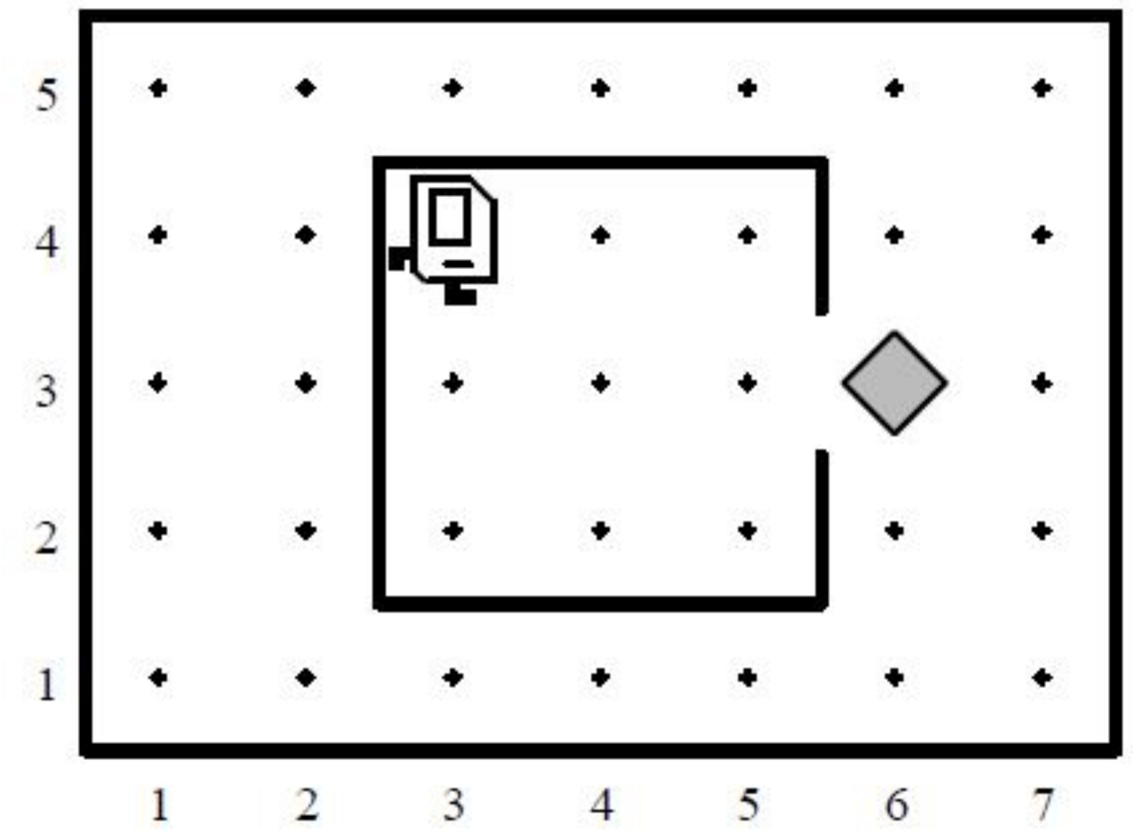
YEAH Hours 2

October 4 2014, 2-3 PM
Nick Troccoli

What are 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

Bye Karel!



Variables

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

```
int x = 2;
char letter = 'a';
boolean isAwesome = true;
```

== is for true/false!

Variable Names

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

Constants

- Not all variables actually change; those that don’t change should be made into constants

- UPPERCASE_WITH_UNDERSCORES

$$\texttt{private static final }\underbrace{\texttt{double}}_{\text{TYPE}}\ \underbrace{\texttt{PI}}_{\text{NAME}}\ \texttt{=}\ \underbrace{\texttt{3.1415}}_{\text{VALUE}}\texttt{;}$$

Control Structures

`for` versus `while`

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

- for loop used for *definite* iteration
- Generally, we know how many times we want to iterate

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

- while loop used for *indefinite* iteration
- Generally, we don’t know how many times to iterate beforehand

Read Until Sentinel

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

  // …rest of body…
}
```

Example: Error Checking

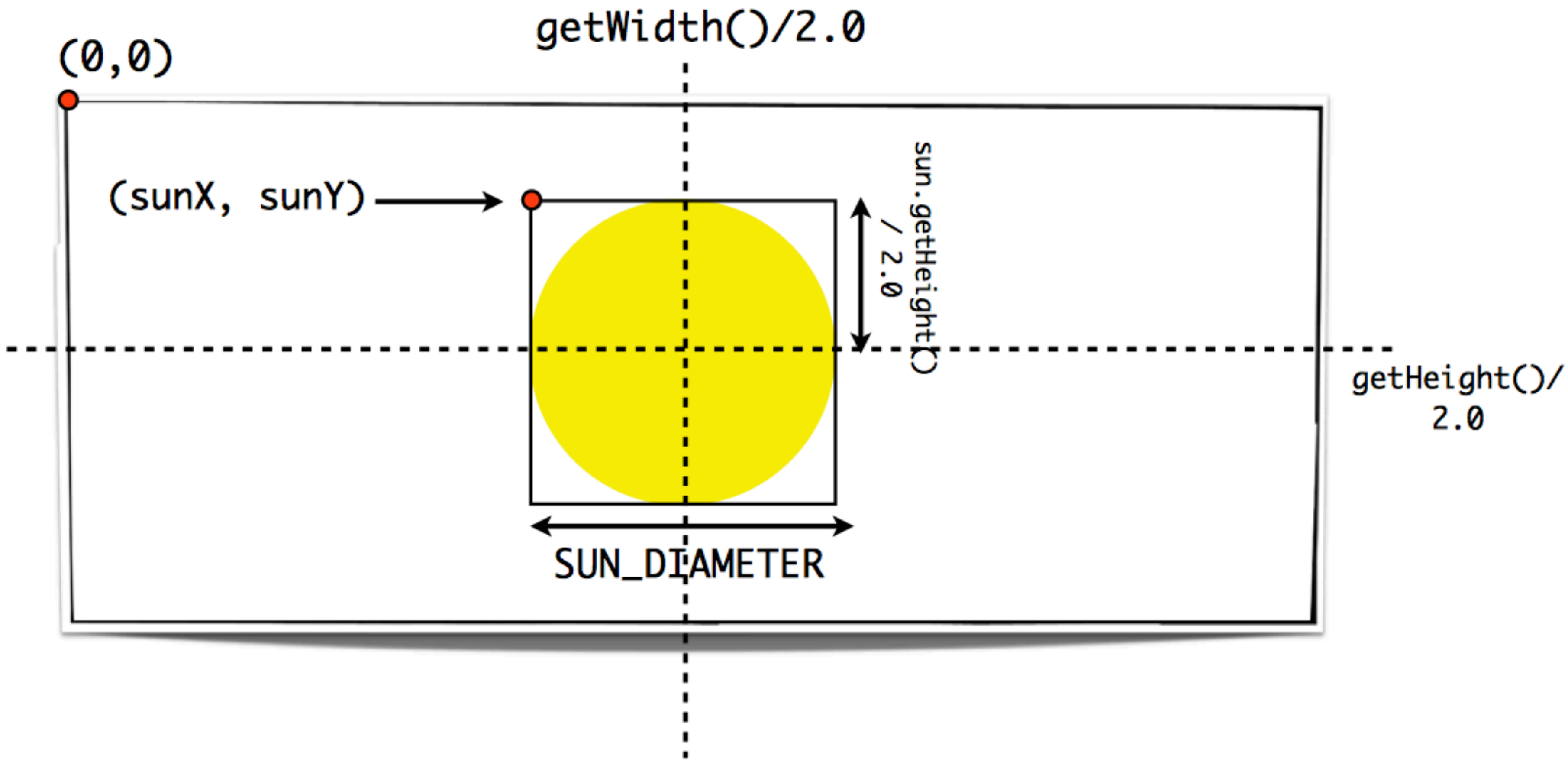
```
int n;

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

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

// use n here (guaranteed positive!)
```

Graphics Warmup

Share

```
private void drawSun() {
    GOval sun = new GOval(SUN_DIAMETER, SUN_DIAMETER);
    sun.setColor(Color.YELLOW);
    sun.setFilled(true);
    sun.setFillColor(Color.YELLOW);
    double sunX = getWidth()/2.0 - sun.getWidth()/2.0;
    double sunY = getHeight()/2.0 - sun.getHeight()/2.0;
    add(sun, sunX, sunY);
}
```

!!



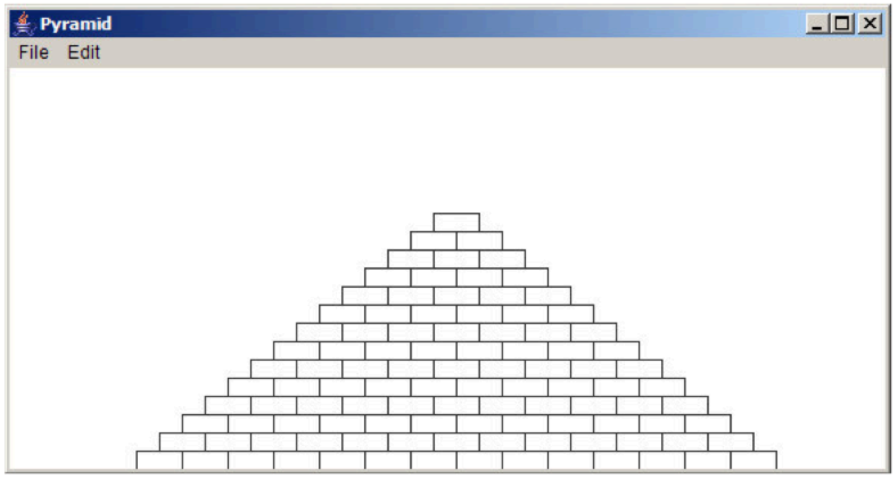
Assignment 2!

Assignment 2: Console & Graphics Programs

- Due Monday October 13 at 3:15PM
- First 3 are console, last 3 are graphics
- No particular order of difficulty

- Key for style: use methods/parameters to decompose

1. Pyramid

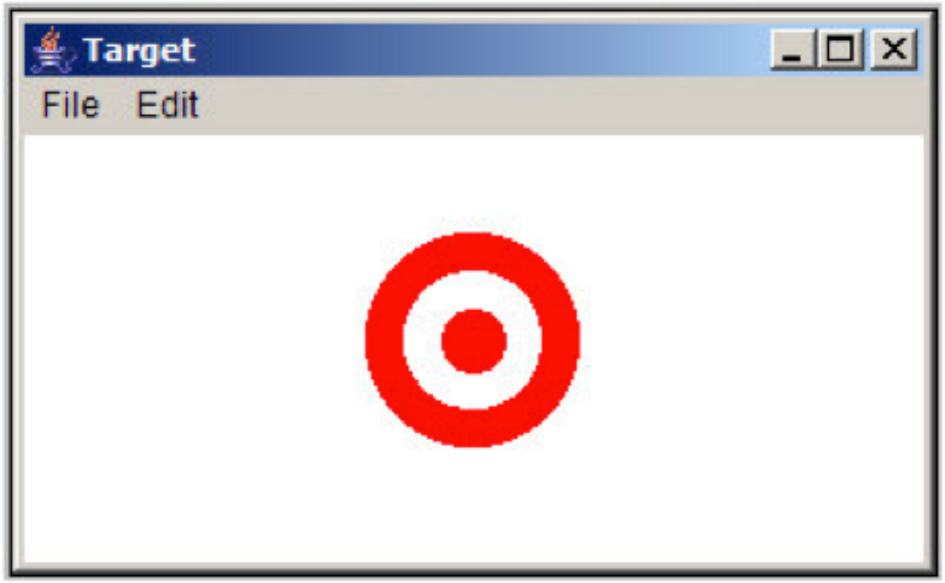


```
/** Width of each brick in pixels */
private static final int BRICK_WIDTH = 30;

/** 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;
```

- Try looking below the window
- <u>Testing:</u> try changing the given constants
- Extensions?

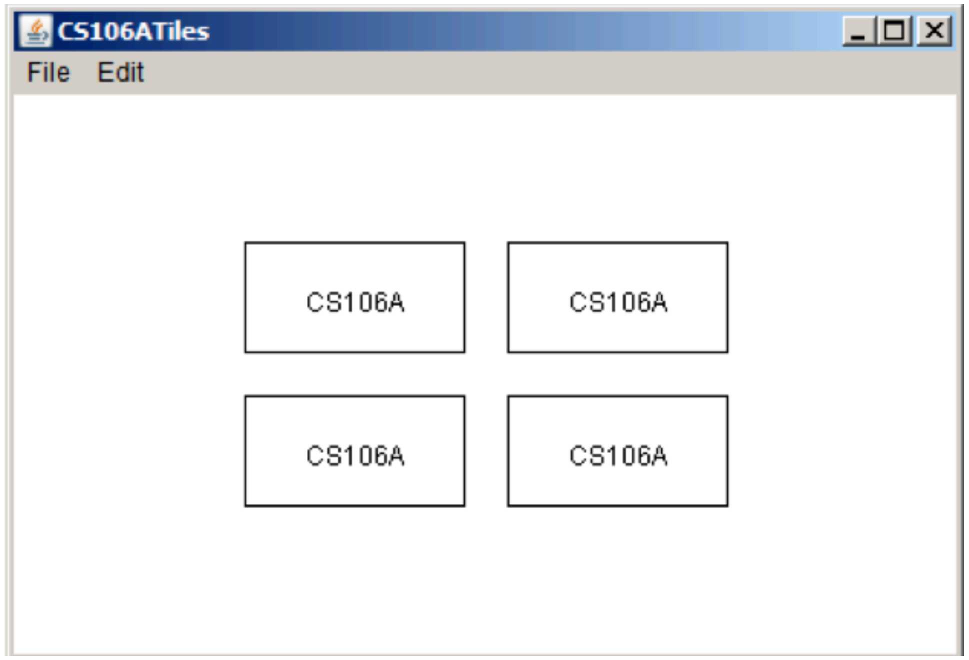
2. Target

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. (from handout)



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

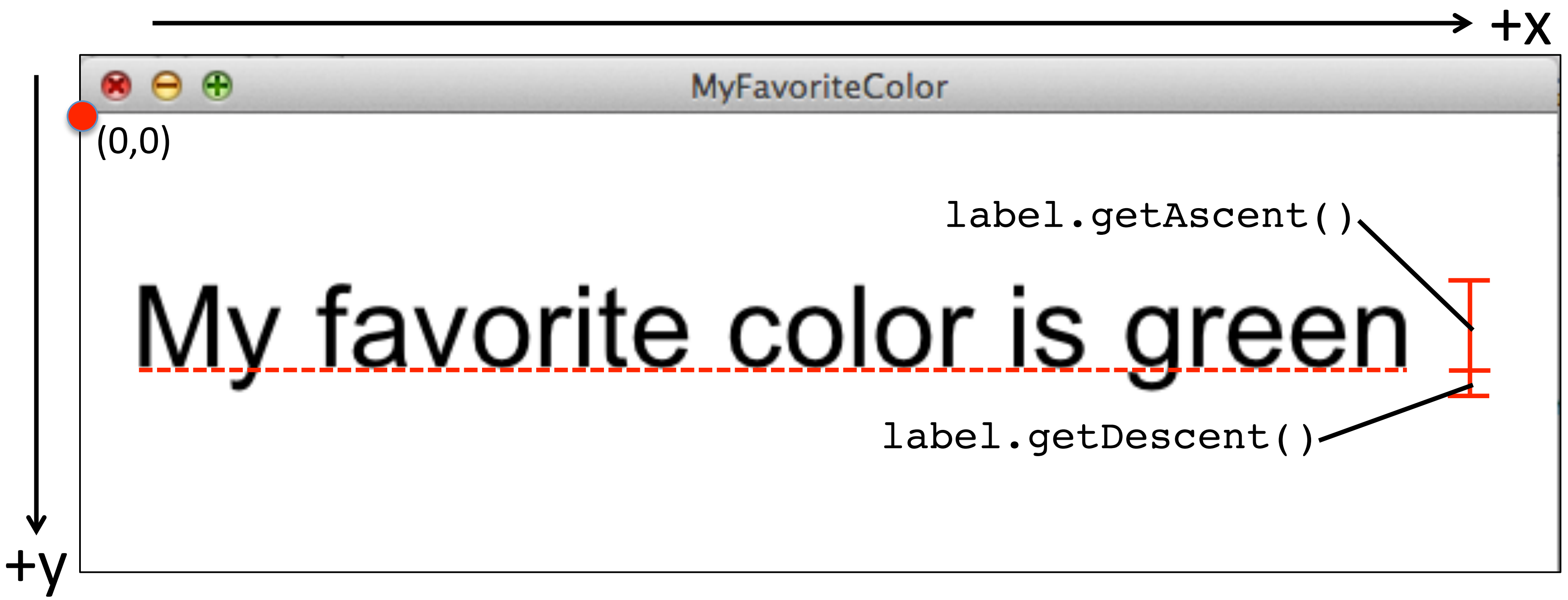
3. CS106A Tiles



```
/** Amount of space (in pixels) between tiles */
private static final int TILE_SPACE = 20;
```

- Think of it as one big rectangle
- TILE_WIDTH, TILE_HEIGHT, TILE_SPACE
- <u>Testing</u>: try changing the given constants
- Centering GLabels

GLabels

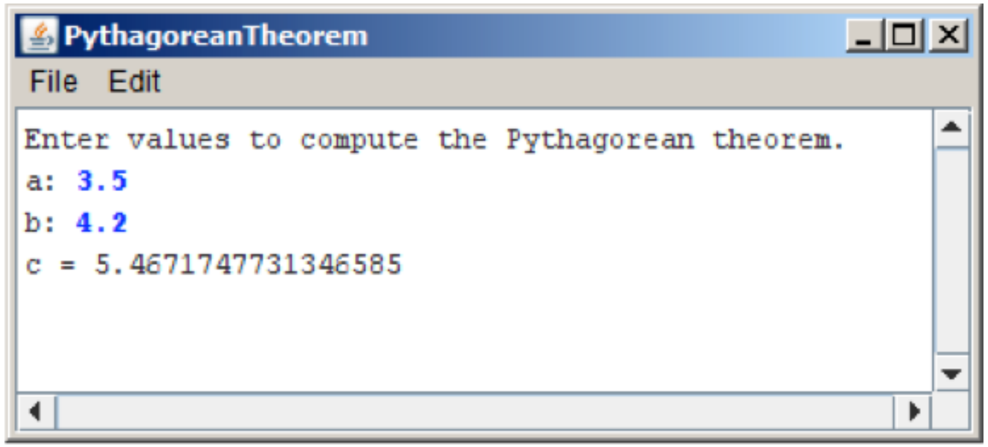


```
GLabel label = new Glabel("My favorite color is green");
// use label.getAscent(), not label.getHeight()!
// (that way label is centered according to baseline)
double x = getWidth()/2.0 – label.getWidth()/2.0;
double y = getHeight()/2.0 – label.getAscent()/2.0;
// label size depends on text and font – 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()

4. Pythagorean Theorem



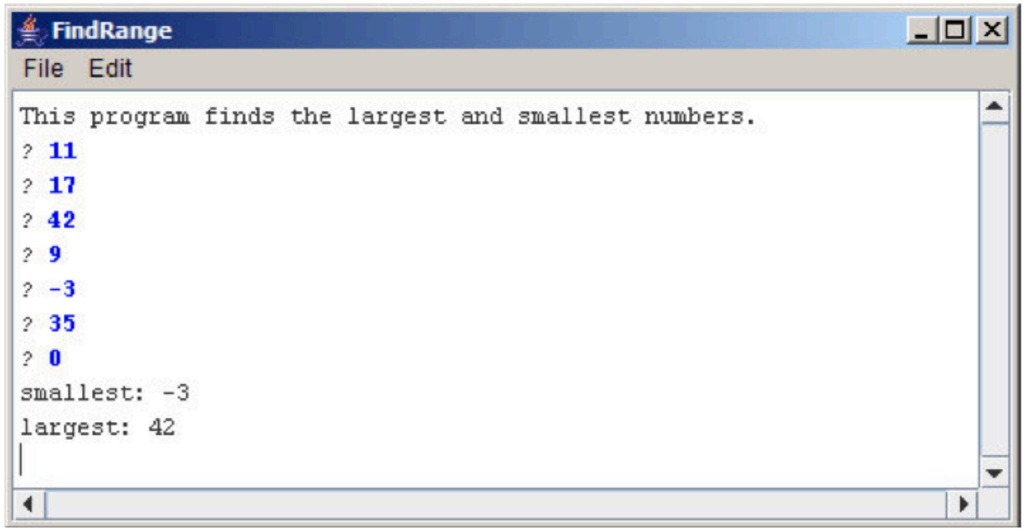
```
Enter values to compute the Pythagorean theorem.
a: 3.5
b: 4.2
c = 5.4671747731346585
```

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

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

- Can assume inputs are positive
- Use **double**!
- Order of operators in Java: * (multiplication), / (division), + (addition), - (subtraction)

5. Max/Min


FindRange

File Edit

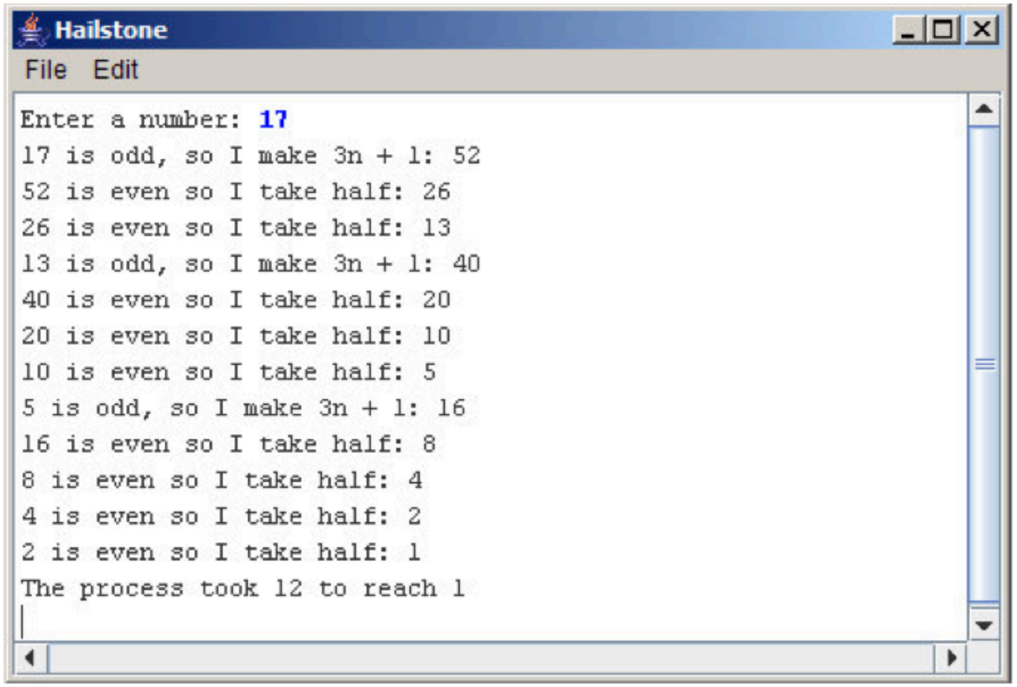
```
This program finds the largest and smallest numbers.
? 11
? 17
? 42
? 9
? -3
? 35
? 0
smallest: -3
largest: 42
```

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)
- <u>Testing:</u> 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
- <u>Testing:</u> 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

Final Tips

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

- Have fun!