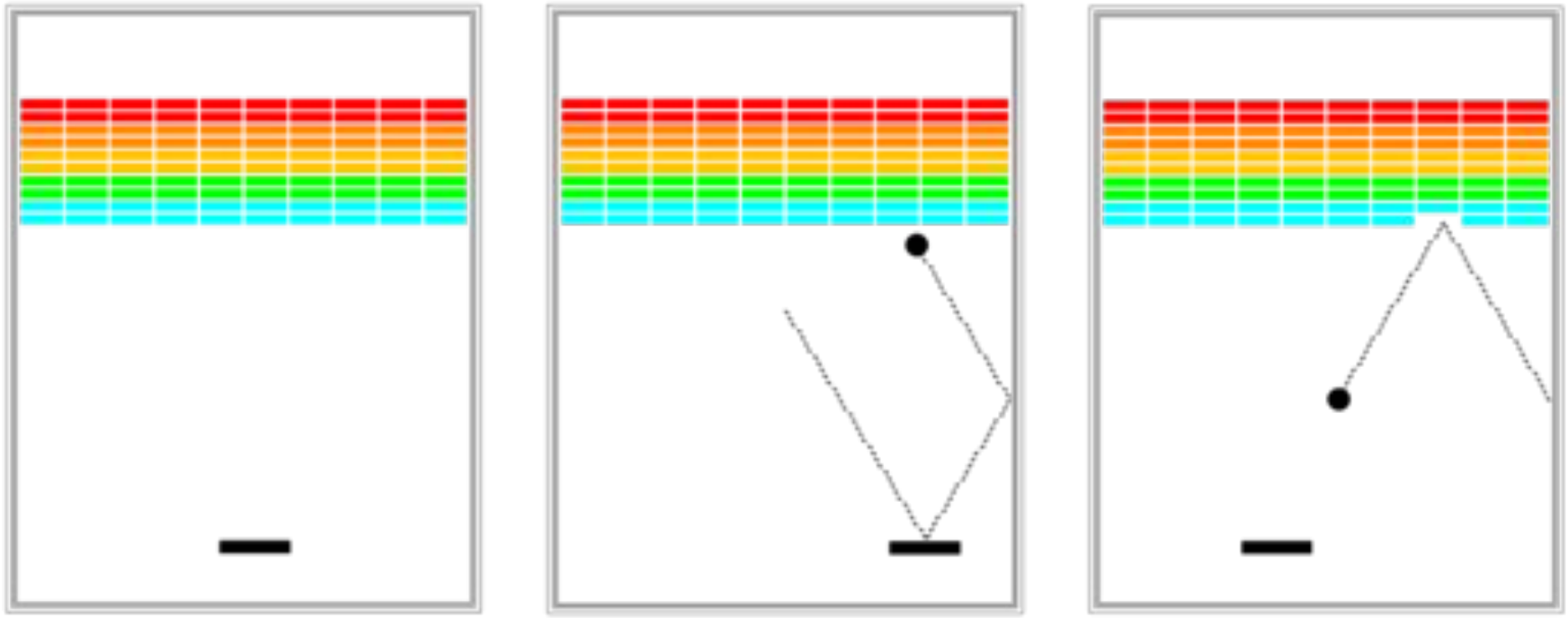
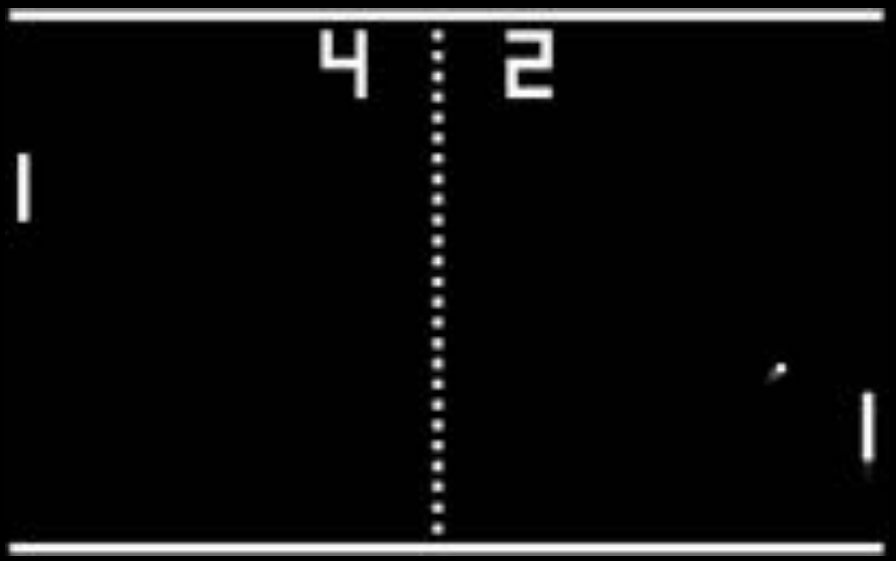
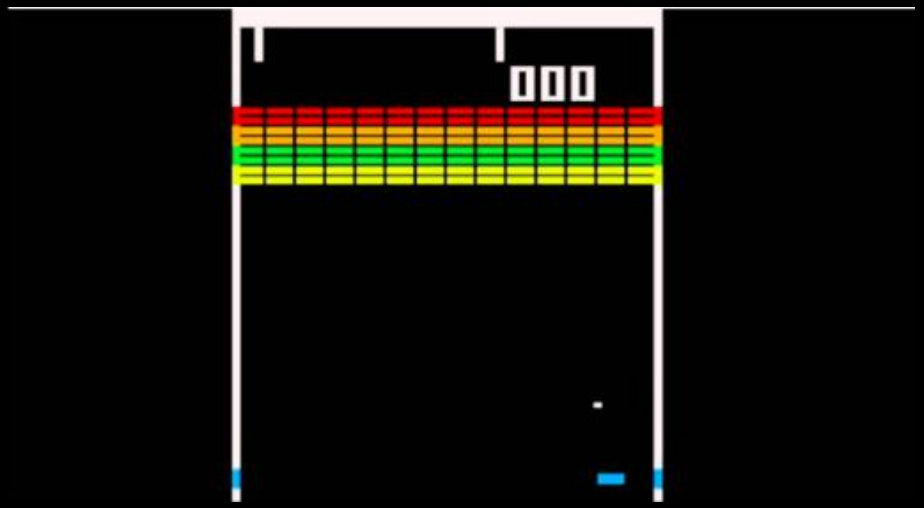


Breakout

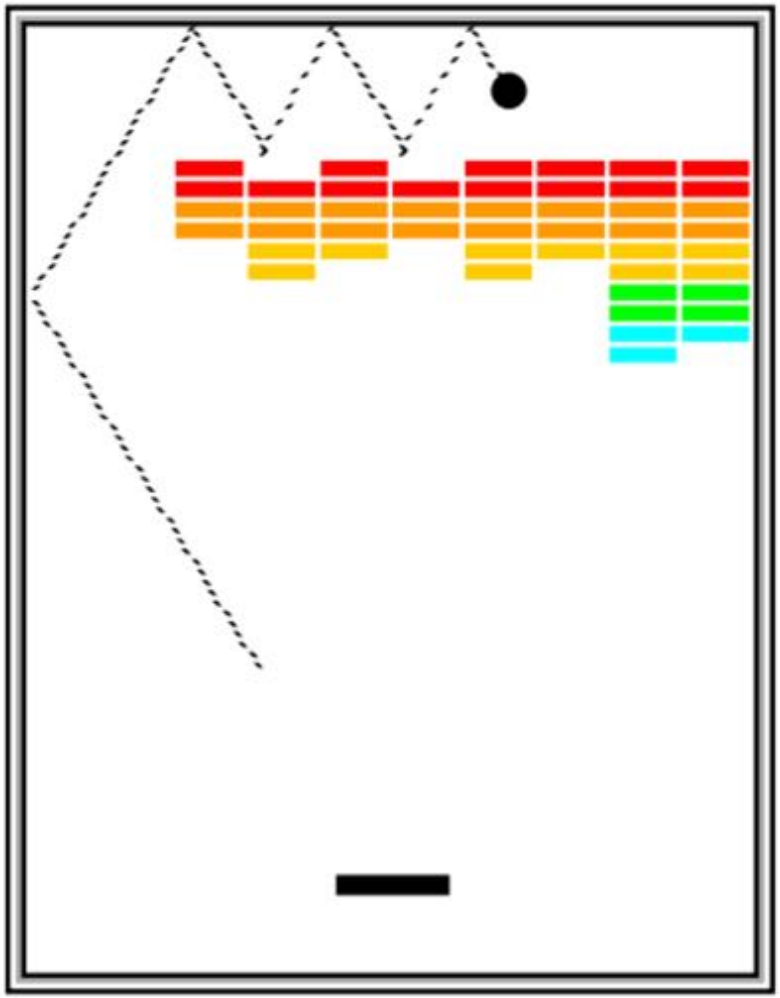
1972



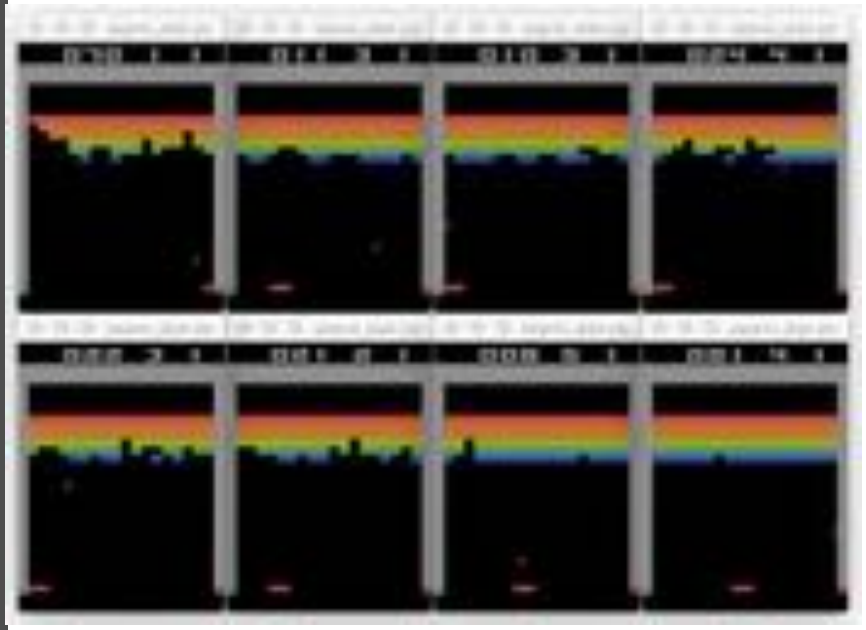


1976


Over 10,000 students
Due in week 5 of CS106A

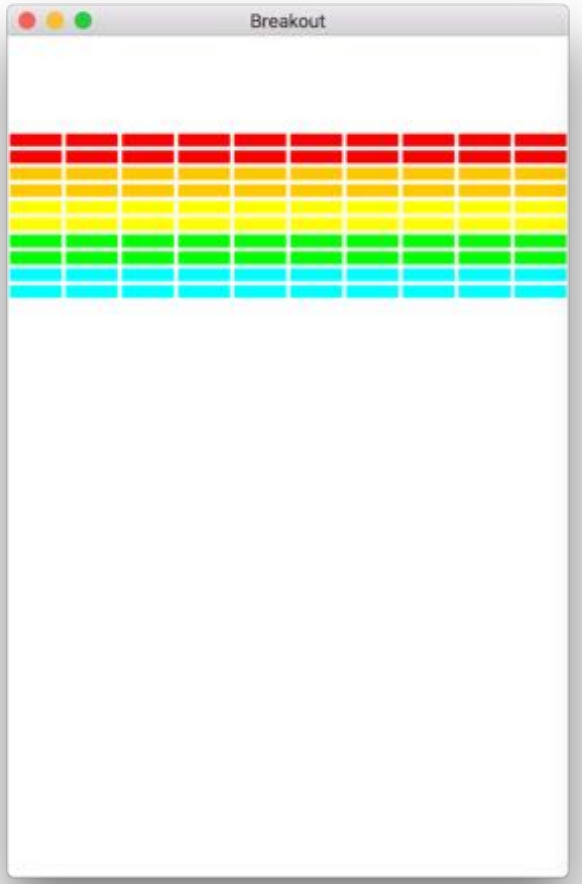



DQN
with tensorflow

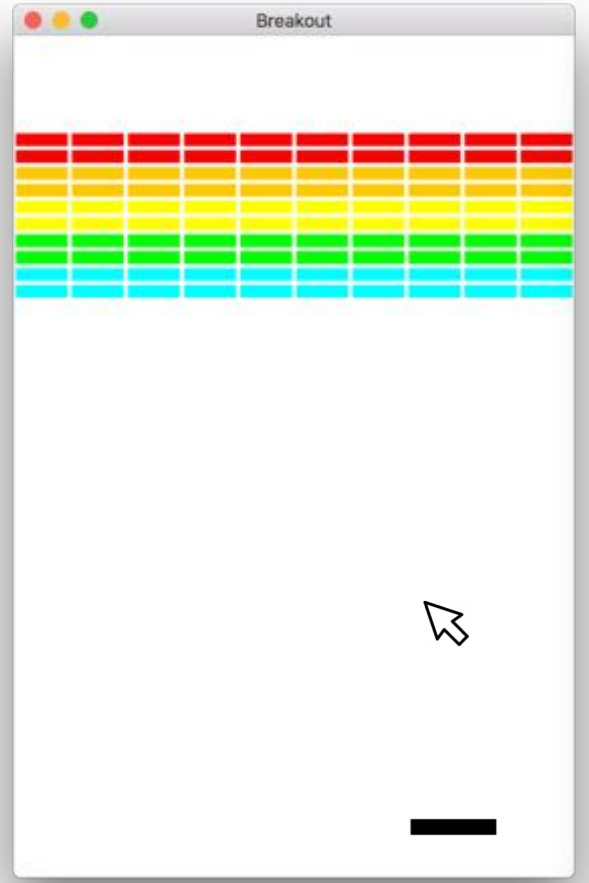


Big program. Do it in parts

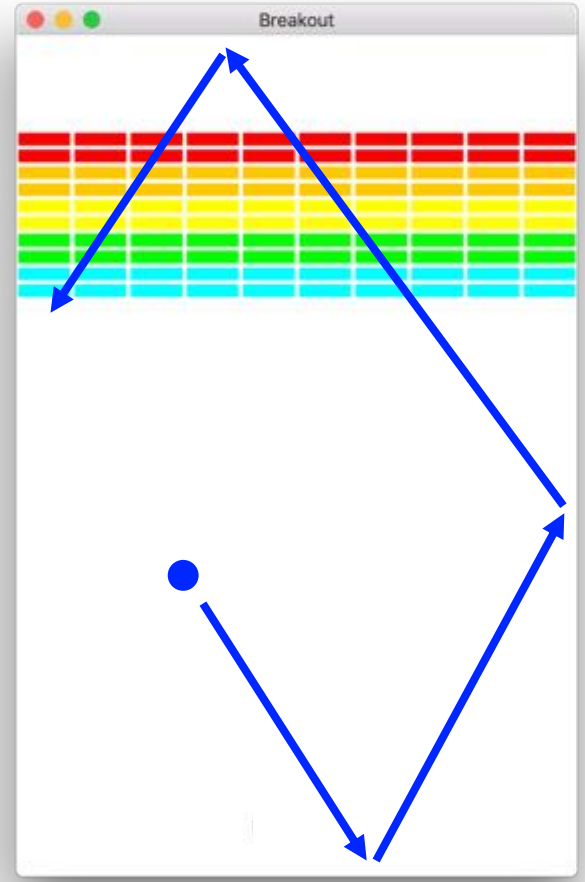
1



2

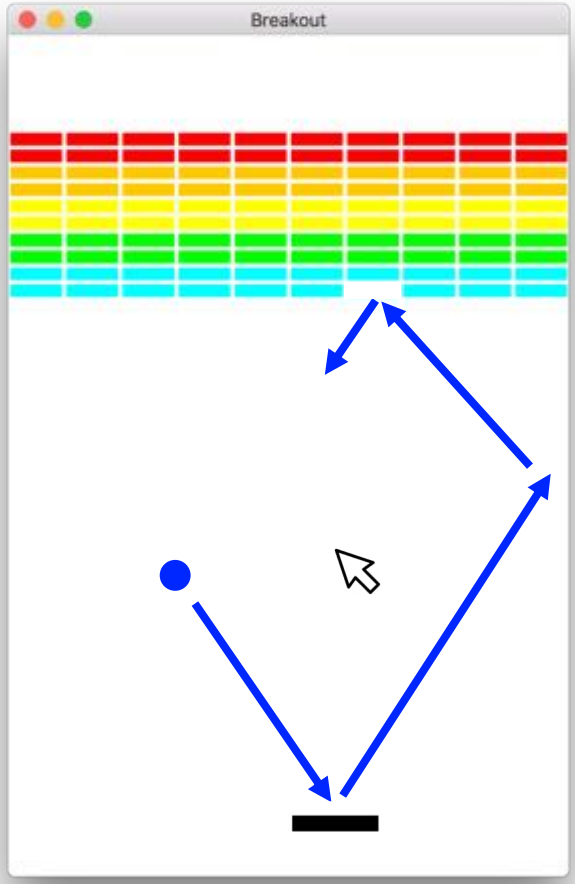


3

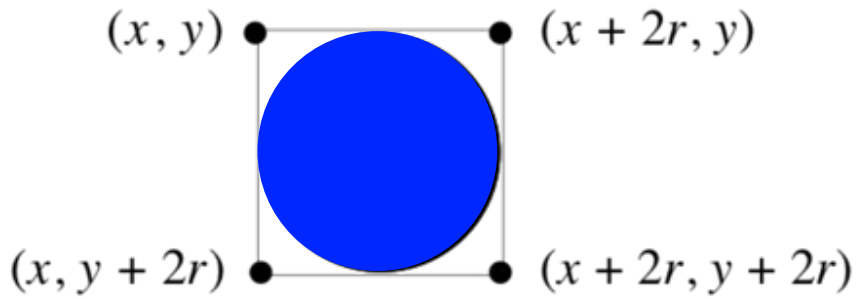


getCollidingObject

4




```
GObject collider =
   getElementAt(x, y);
```



`null` ,  , *a brick*

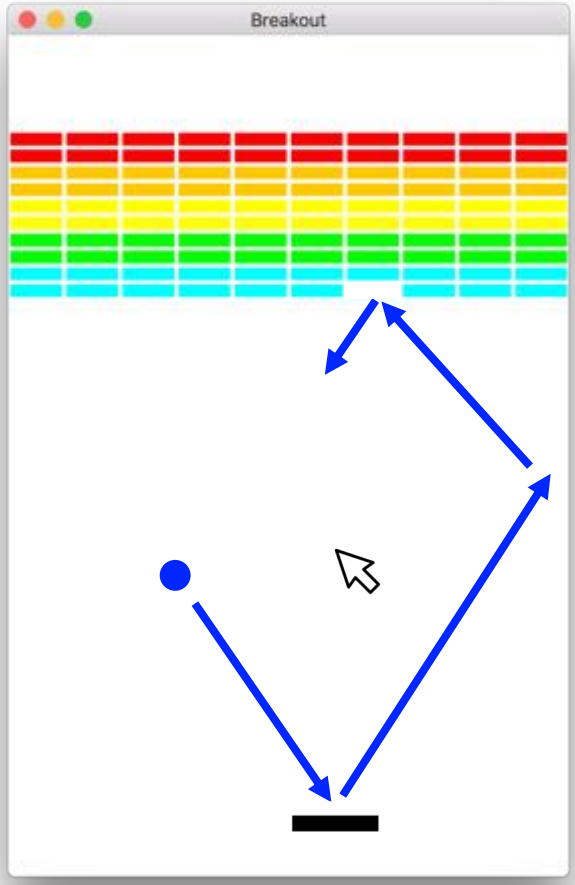

Fuge
Ferris Wheel

Pro Tips

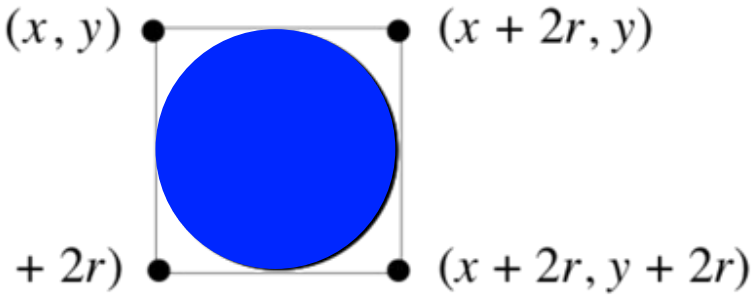
- Want to wait for a click to start? Use **waitForClick()**
- Do not animate in **mouse moved**!
- Use **instance** var for paddle.
- Make sure to **test as you go**. Program one milestone at a time.
- No instance variable **for bricks**

How do you know if you hit a brick?

4




```
GObject collider =
   getElementAt(x, y);
```



`null` ,  , *a brick*

Aside: Secret to how memory really works

Who thinks this prints **`true`**?

```
public void run() {
      int x = 5;
      int y = 5;
      println(x == y);
}
```

Who thinks this prints **`true`**?

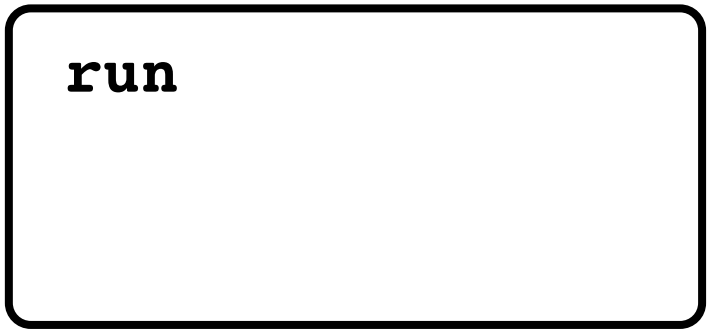
```
public void run() {
    GRect first = new GRect(20, 30);
    GRect second = new GRect(20, 30);
    println(first == second);
}
```

Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```

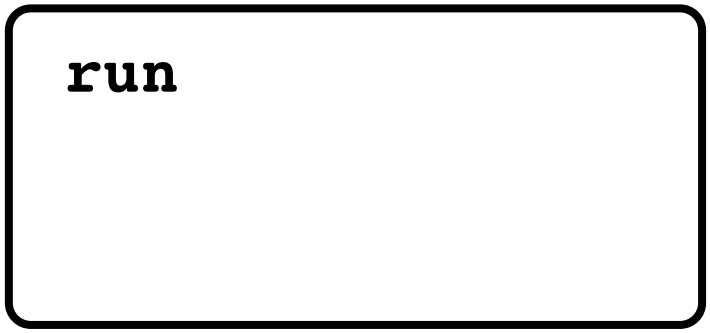
Stack Diagrams

```
public void run() {
    println(toInches(5));
}

private int toInches(int feet){
    int result = feet * 12;
    return result;
}
```



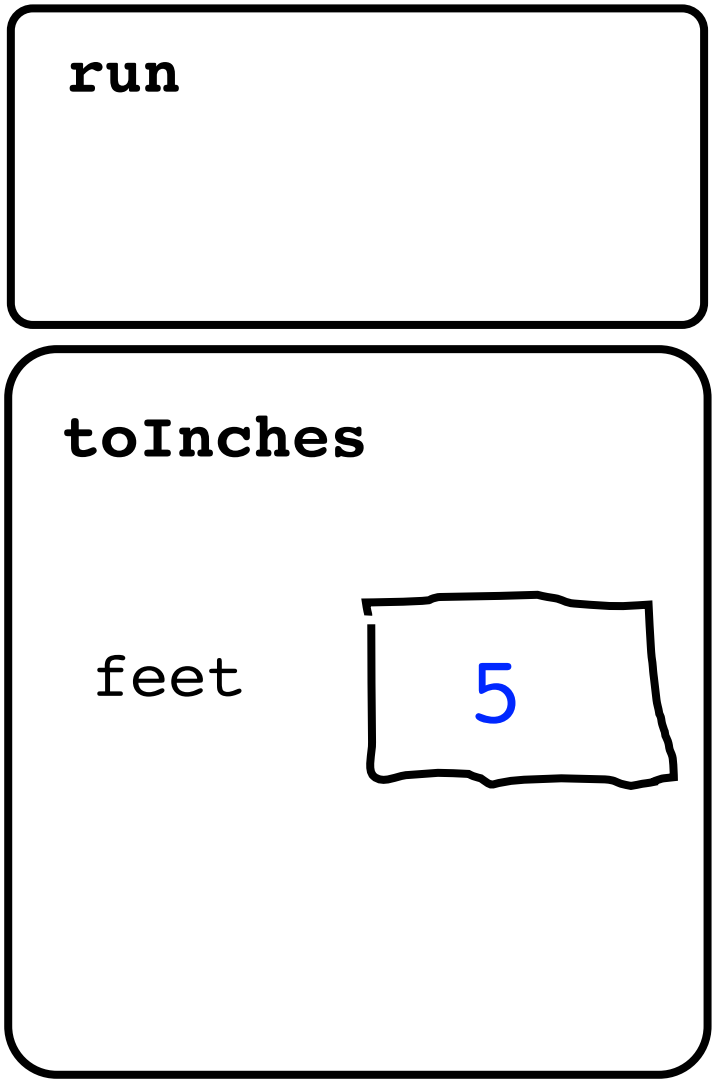
Stack Diagrams

```
public void run() {
    println(toInches(5));
}

private int toInches(int feet){
    int result = feet * 12;
    return result;
}
```



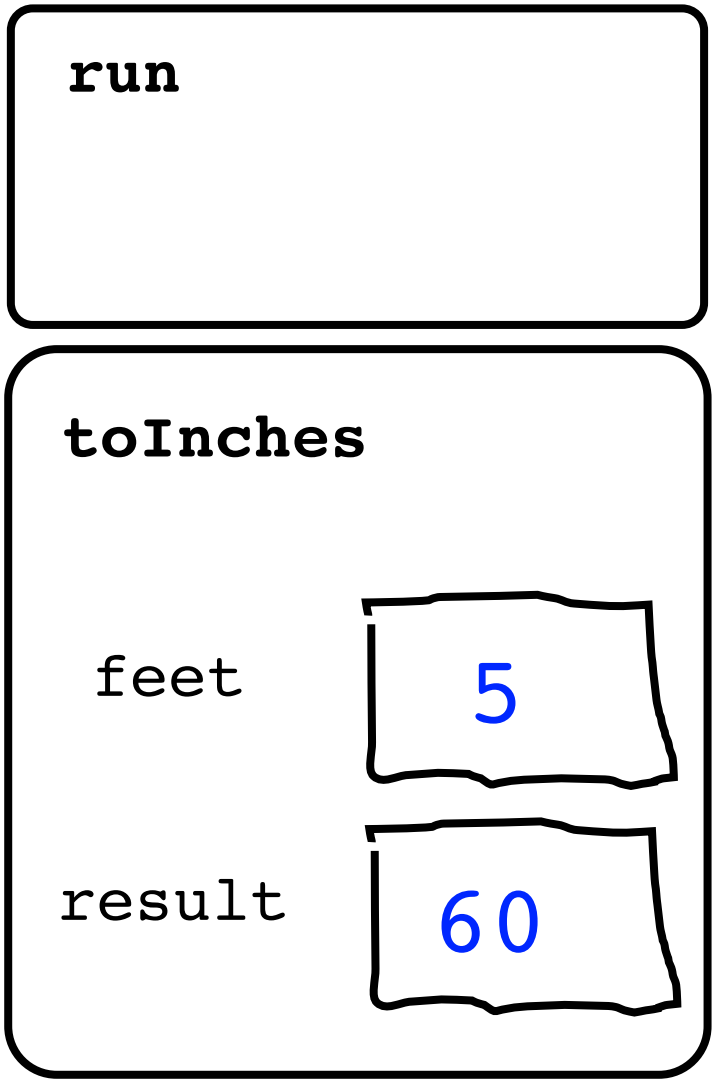
Stack Diagrams

```
public void run() {
    println(toInches(5));
}

private int toInches(int feet){
    int result = feet * 12;
    return result;
}
```



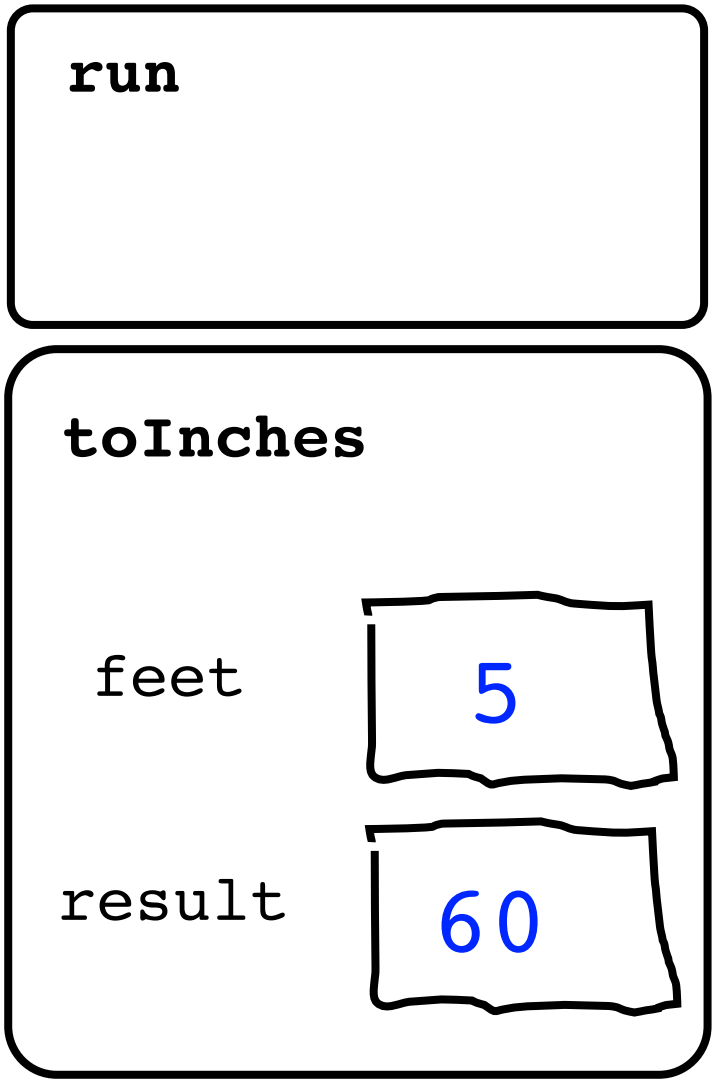
Stack Diagrams

```
public void run() {
    println(toInches(5));
}

private int toInches(int feet){
    int result = feet * 12;
    return result;
}
f
```



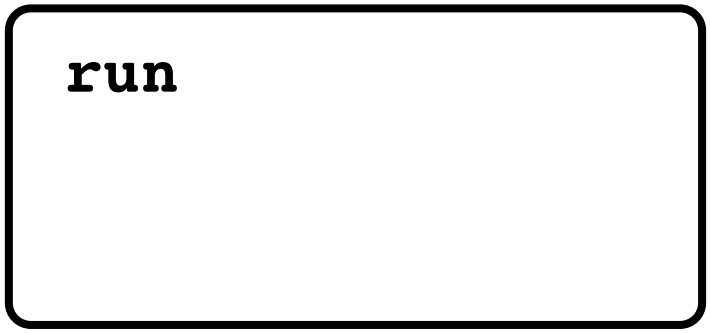
Stack Diagrams

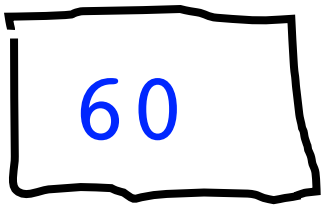
```
public void run() {
    println(toInches(5));
}

private int toInches(int feet){
    int result = feet * 12;
    return result;
}
```



Stack Diagrams

```
public void run() {
    println(toInches(5));
}

private int toInches(int feet){
    int result = feet * 12;
    return result;
}
```

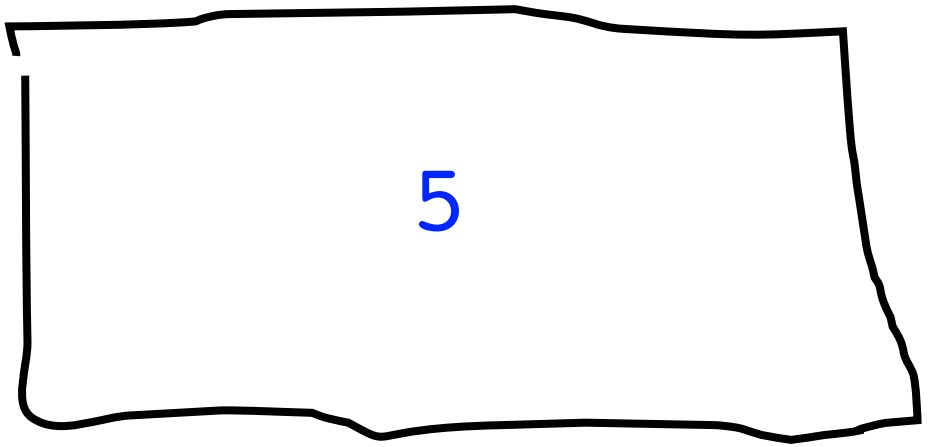




Aside: Actual Memory

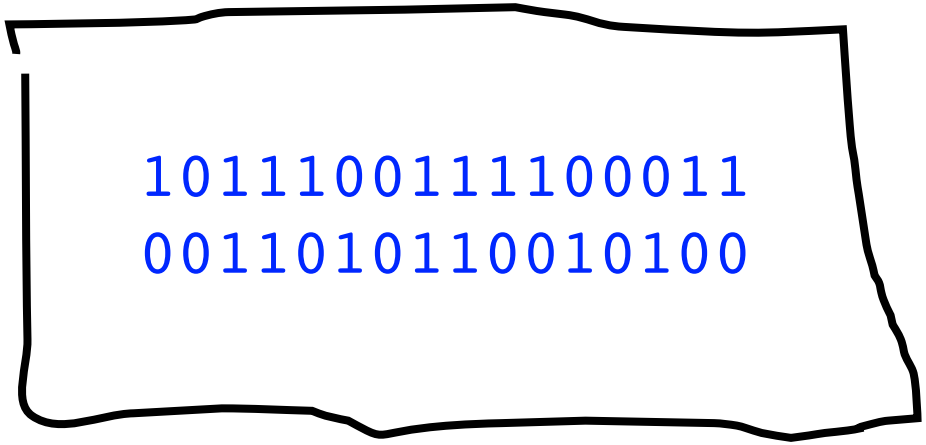
What is a bucket

feet



What is a bucket

feet



* Each bucket or “word” holds 64 bits

** don’t think on the binary level (yet)

variables have fixed size buckets to store values

Primitives vs Classes

Primitive Variable Types

```
int
double
char
boolean
```

Class Variable Types

```
GRect
GOval
Gline
Color
```

Class variables (aka objects)

1. Have upper camel case types
2. You can call methods on them
3. Are constructed using `new`
4. Are stored in a special way

Primitives vs Classes

Primitive Variable Types

```
int
double
char
boolean
```

Class Variable Types

```
GRect
GOval
Gline
Color
```

aka Objects

Class variables (aka objects)

1. Have upper camel case types
2. You can call methods on them
3. Are constructed using `new`
4. Are stored in a special way

How do you share wikipedia articles?

Antelope Canyon Article

Antelope Canyon is a slot canyon in the American Southwest. It is located on Navajo land east of Page, Arizona. Antelope Canyon includes two separate, photogenic slot canyon sections, referred to individually as *Upper Antelope Canyon* or *The Crack*; and *Antelope Canyon* or *The Corkscrew*.[2]

The Navajo name for Upper Antelope Canyon is Tsé bighánílíní, which means "the place where water runs through rocks." Lower Antelope Canyon is Hazdistazí (advertised as *"Hasdestwazi"* by the Navajo Parks and Recreation Department), or "spiral rock arches." Both are located within the LeChee Chapter of the Navajo Nation.[4]

Contents [hide]

1 Geology
2 Tourism and photography
2.1 Upper Antelope Canyon

Antelope Canyon

Tsé bighánílíní dóó Hazdistazí (Navajo)

A beam of light in Upper Antelope Canyon

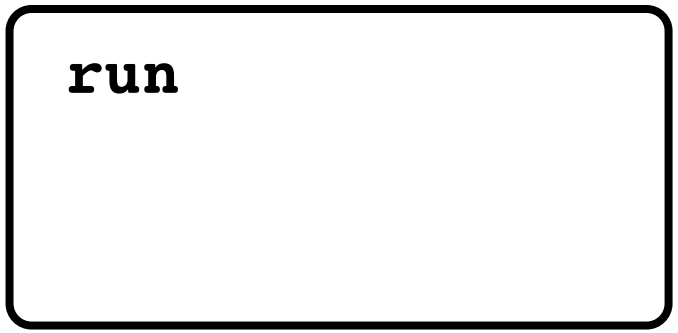
https://en.wikipedia.org/wiki/Antelope_Canyon

```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```

---

stack

heap

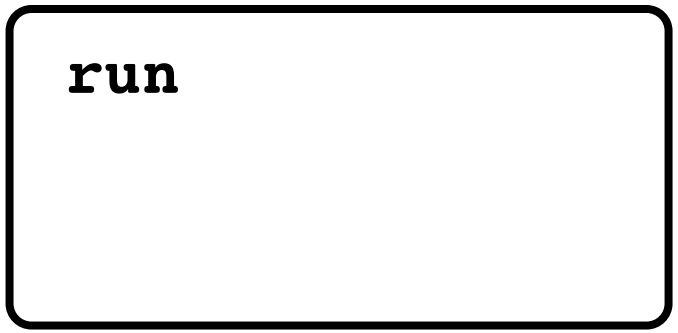


```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```

---

stack

heap

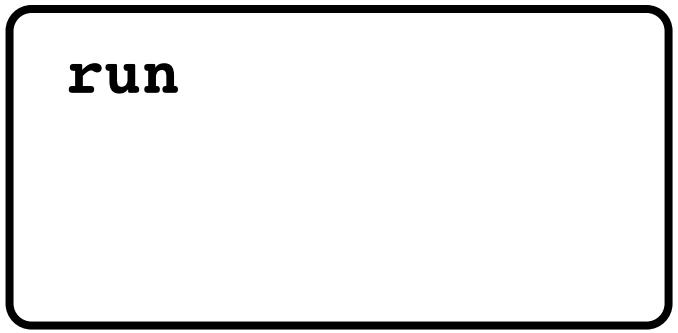


```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```

---

stack

heap

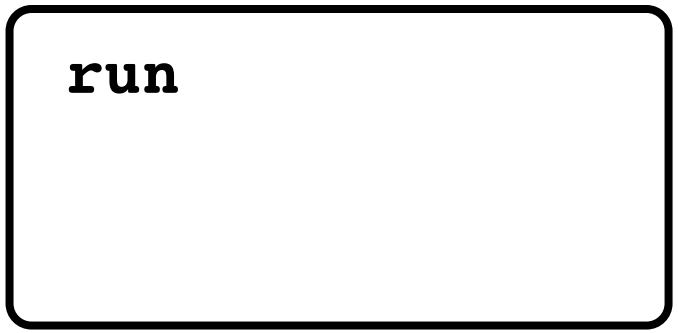


```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```

---

stack

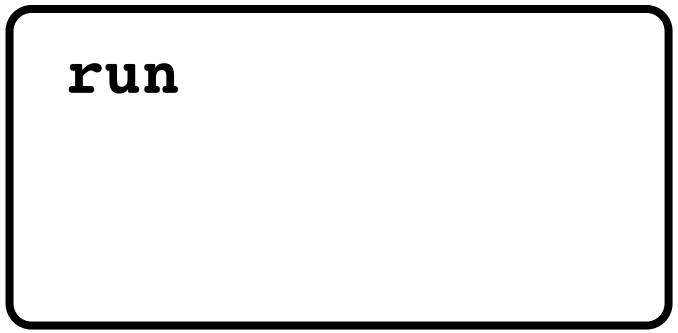
heap



```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```

---

stack



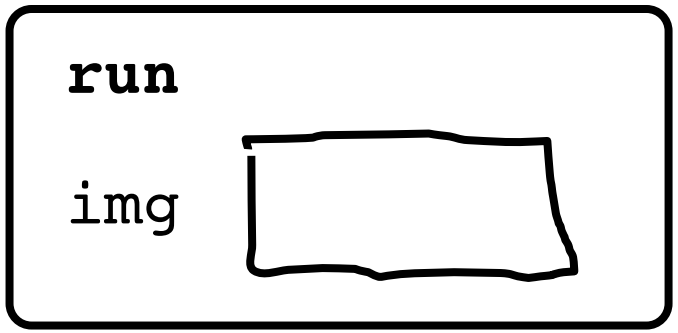
heap

42

```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```

---

stack



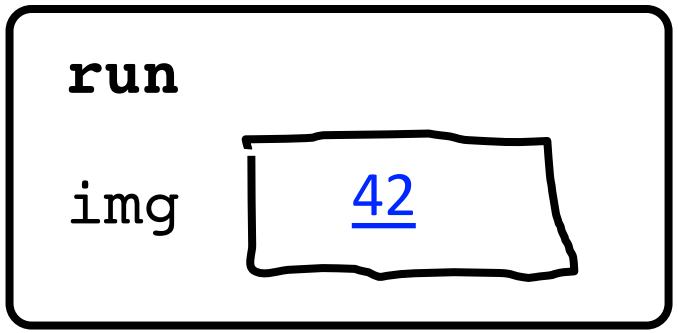
heap

42

```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```

---

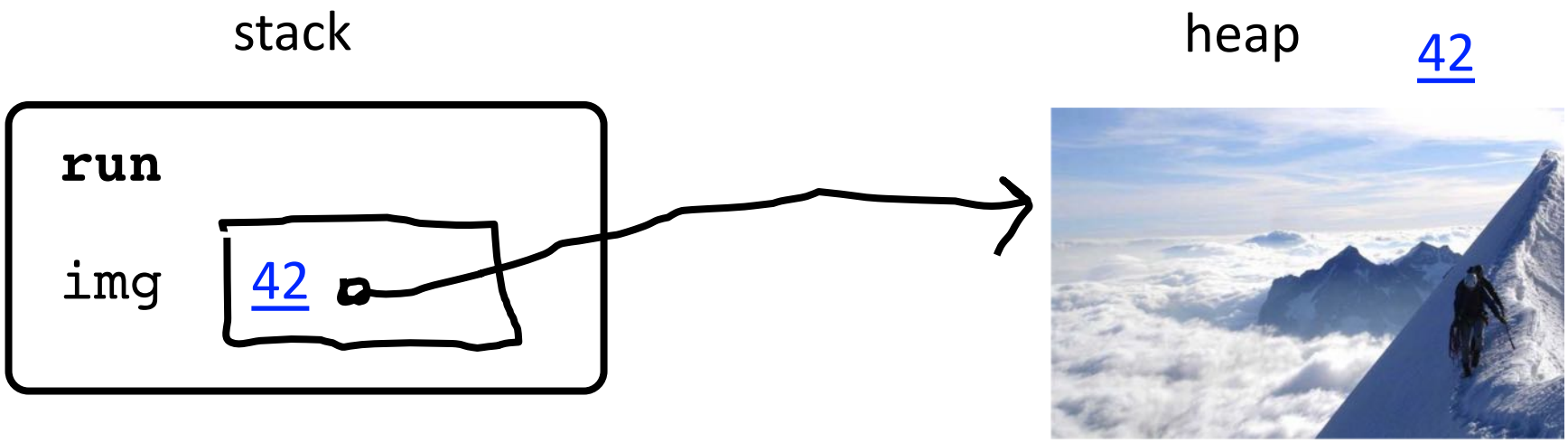
stack



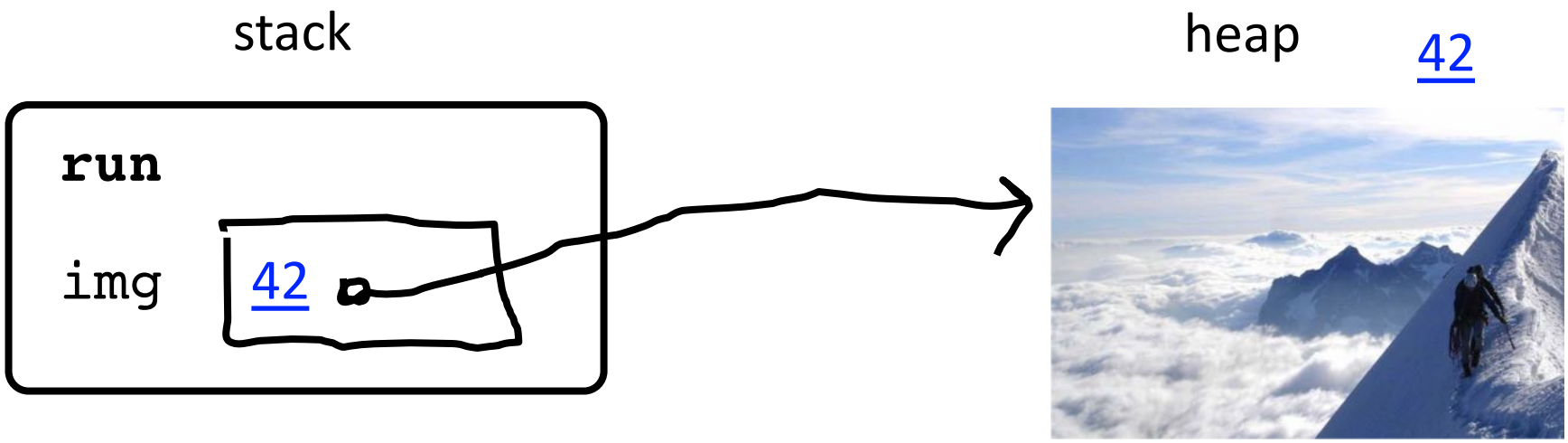
heap

42

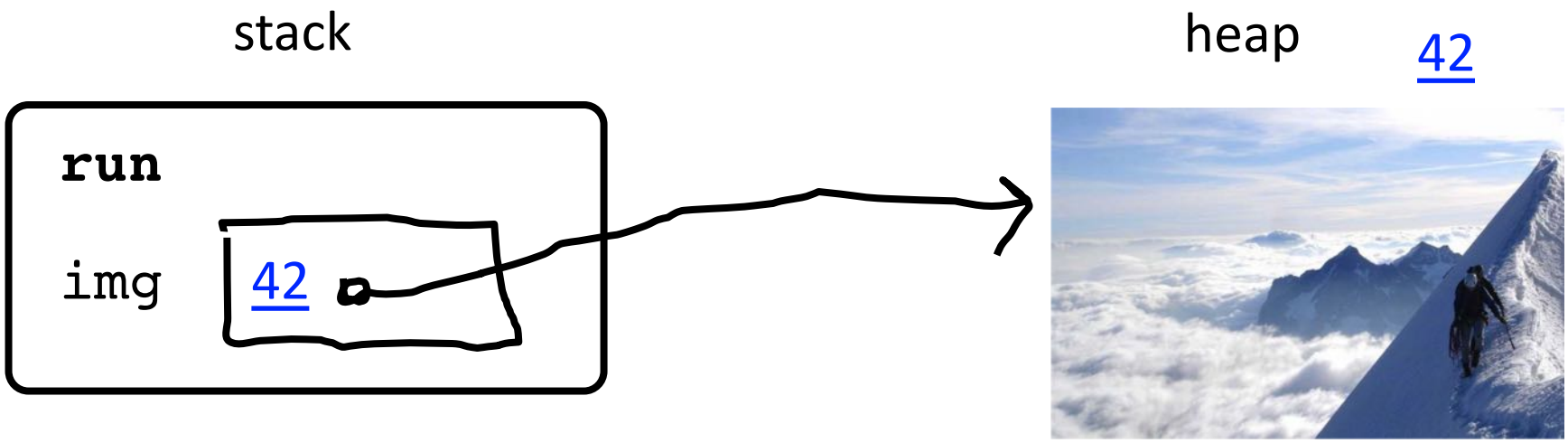
```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```



```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```



```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```



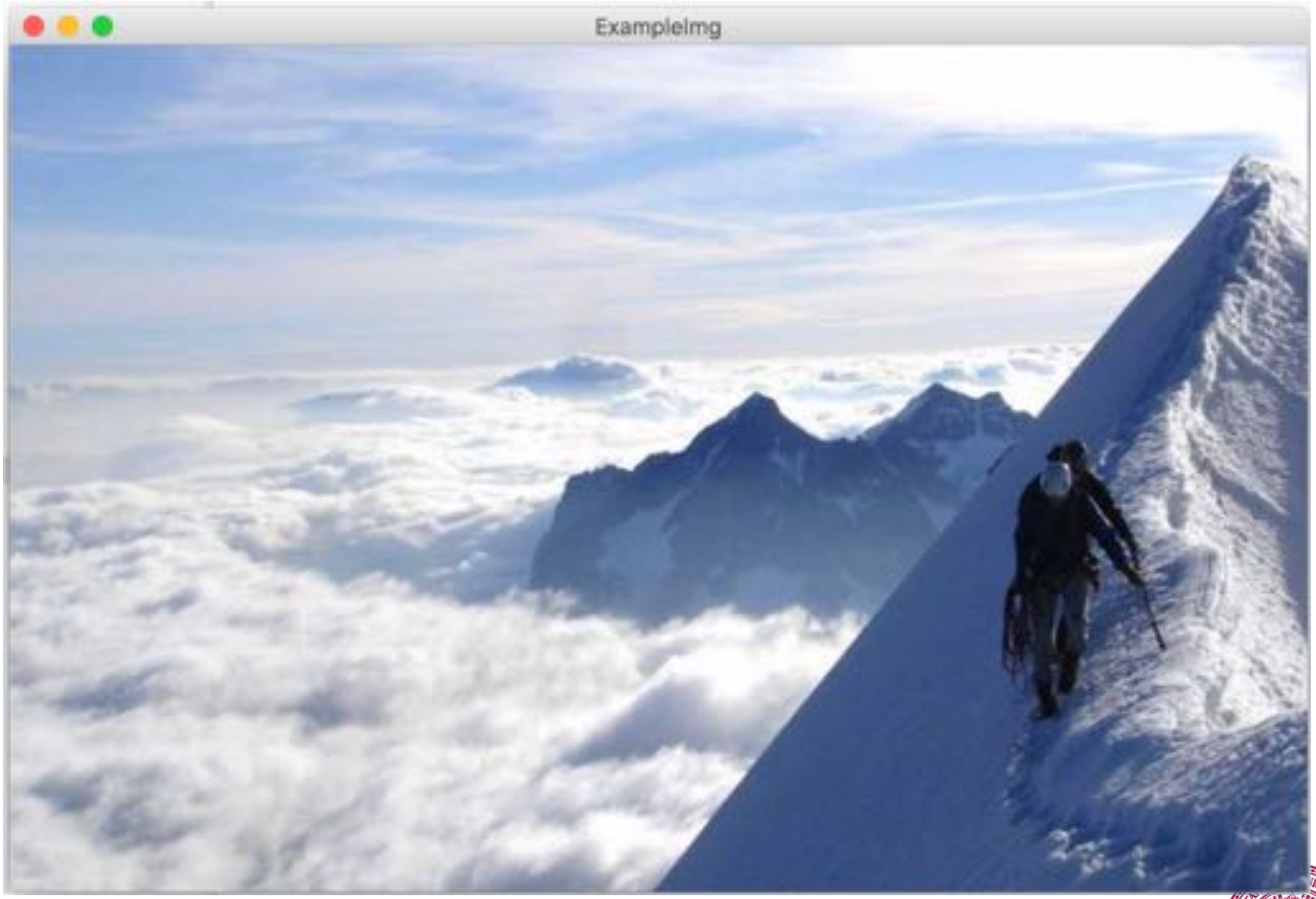
```
public void run() {
    GImage img = new GImage("mountain.jpg");
    add(img, 0, 0);
}
```

---

stack

heap

42


ExampleImg

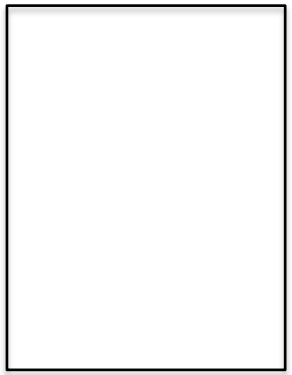
Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```


Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```

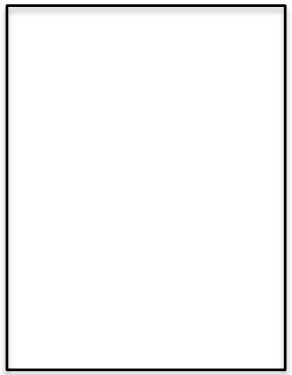
memory.com/18



Who thinks this prints **`true`**?

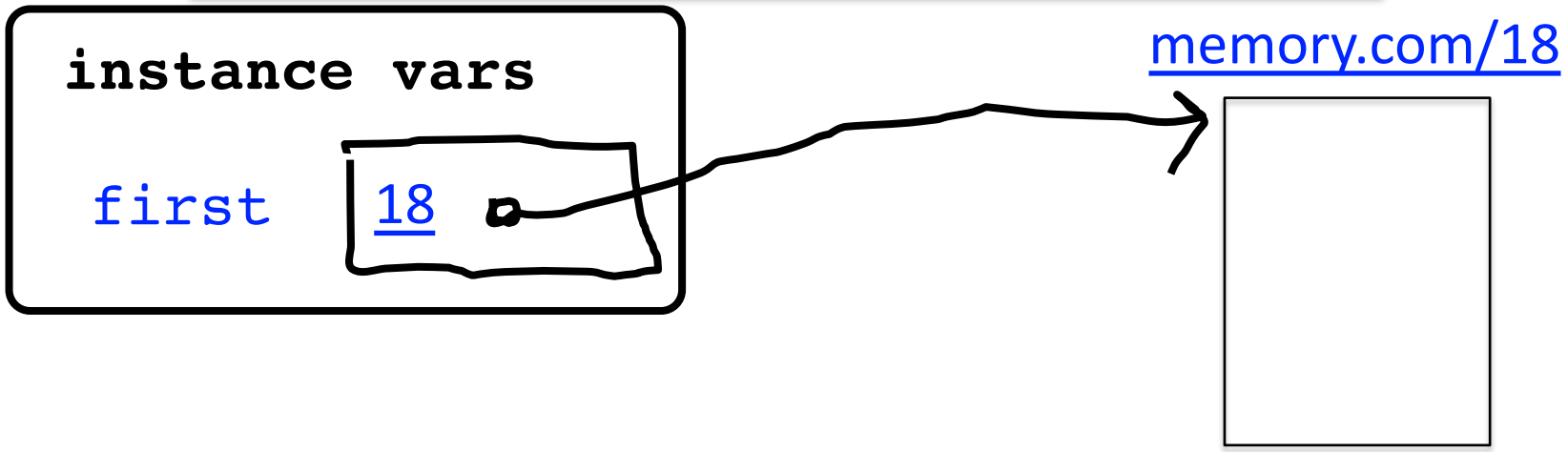
```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```

memory.com/18



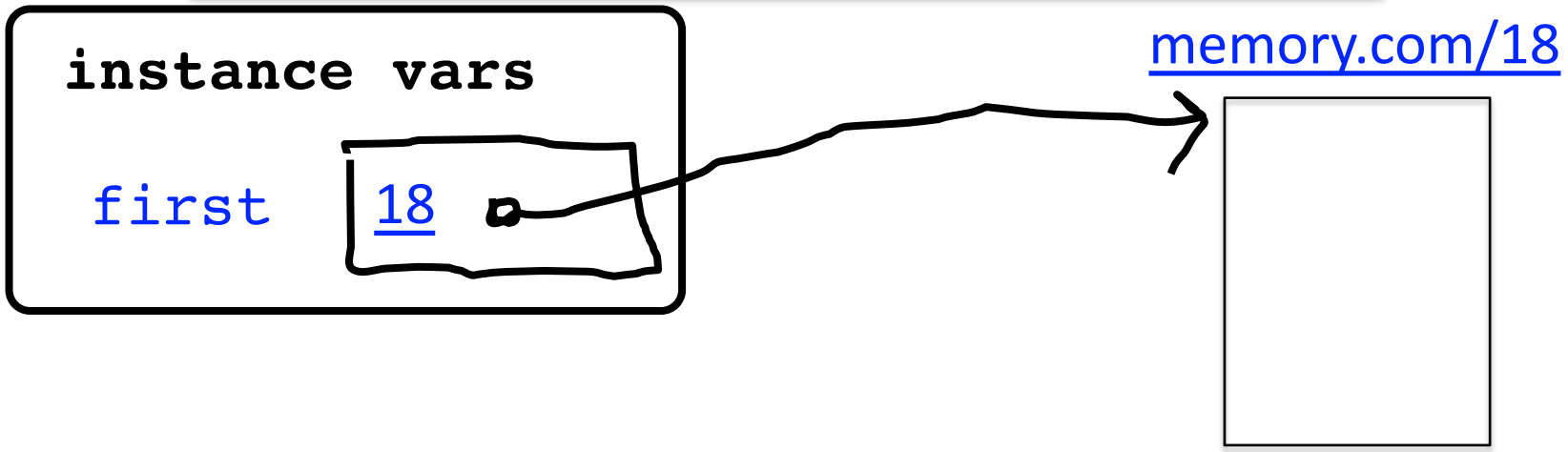
Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```



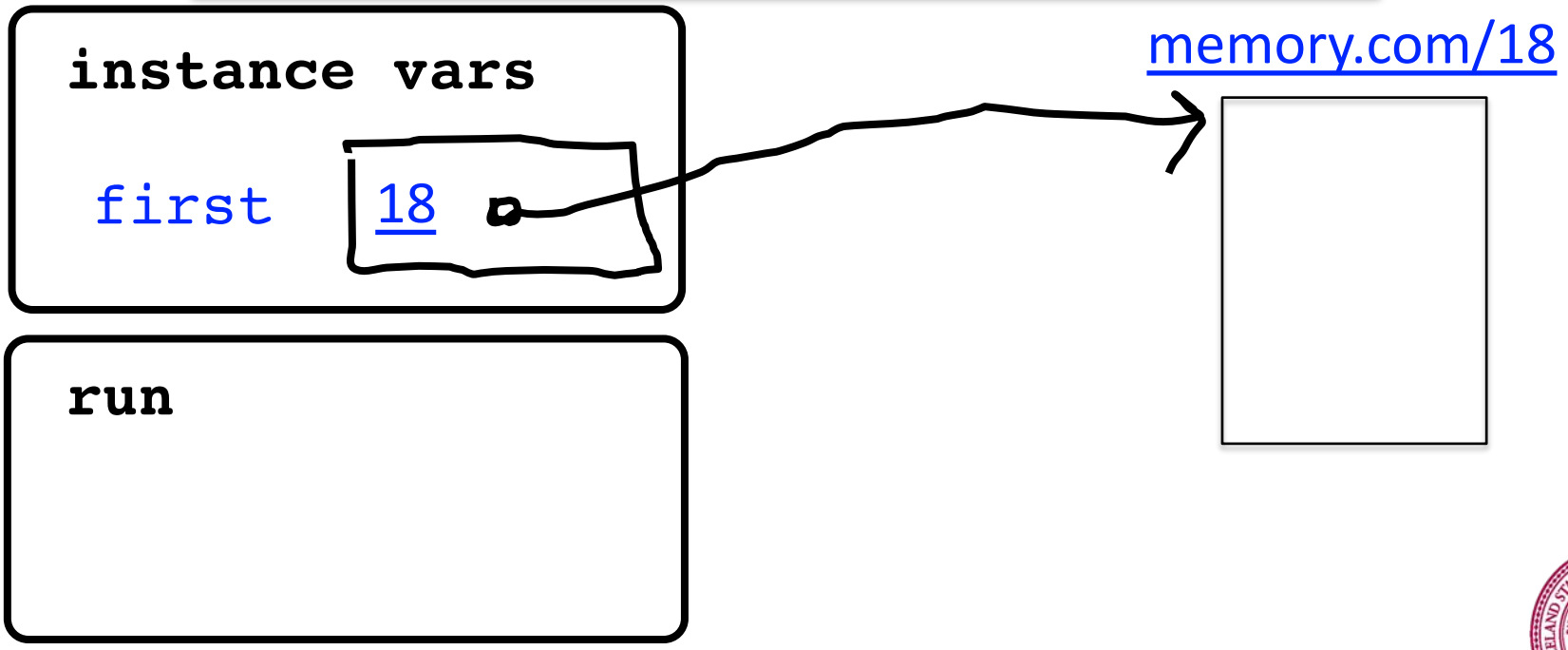
Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```



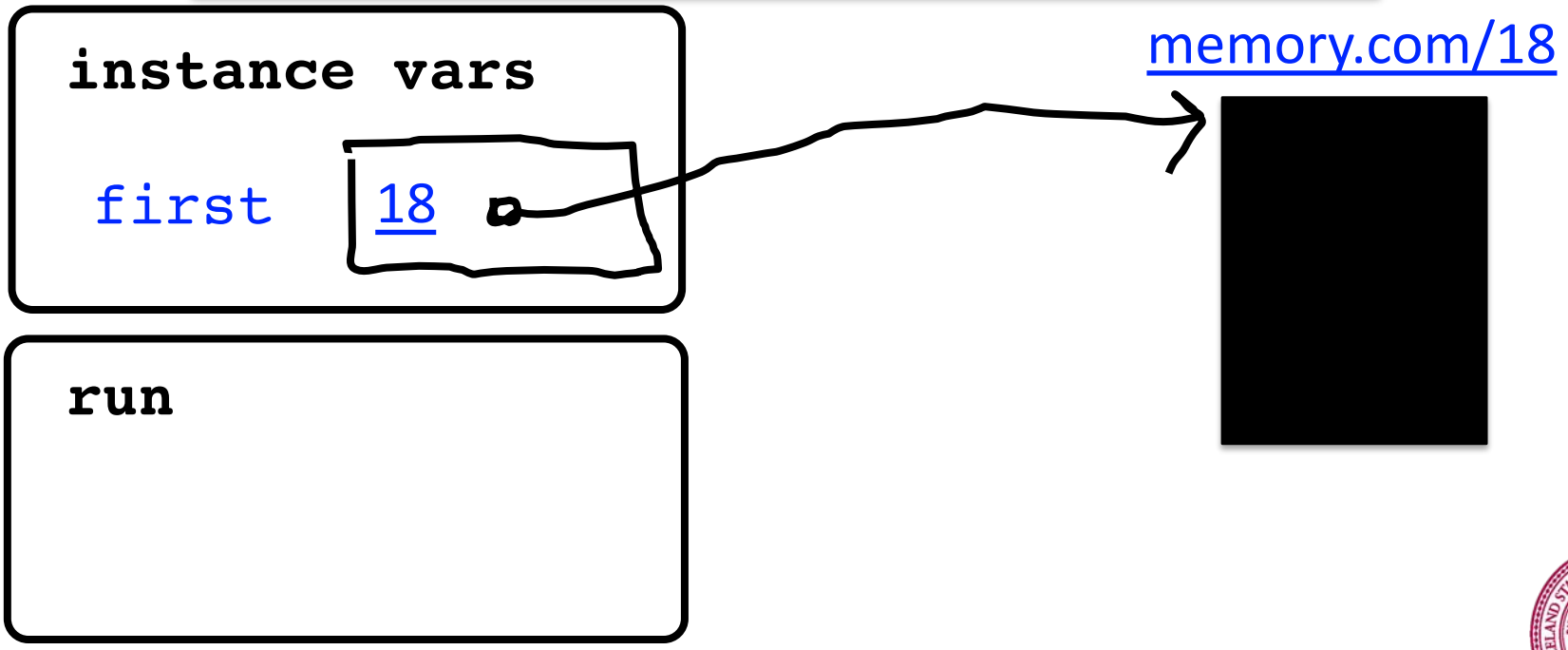
Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```



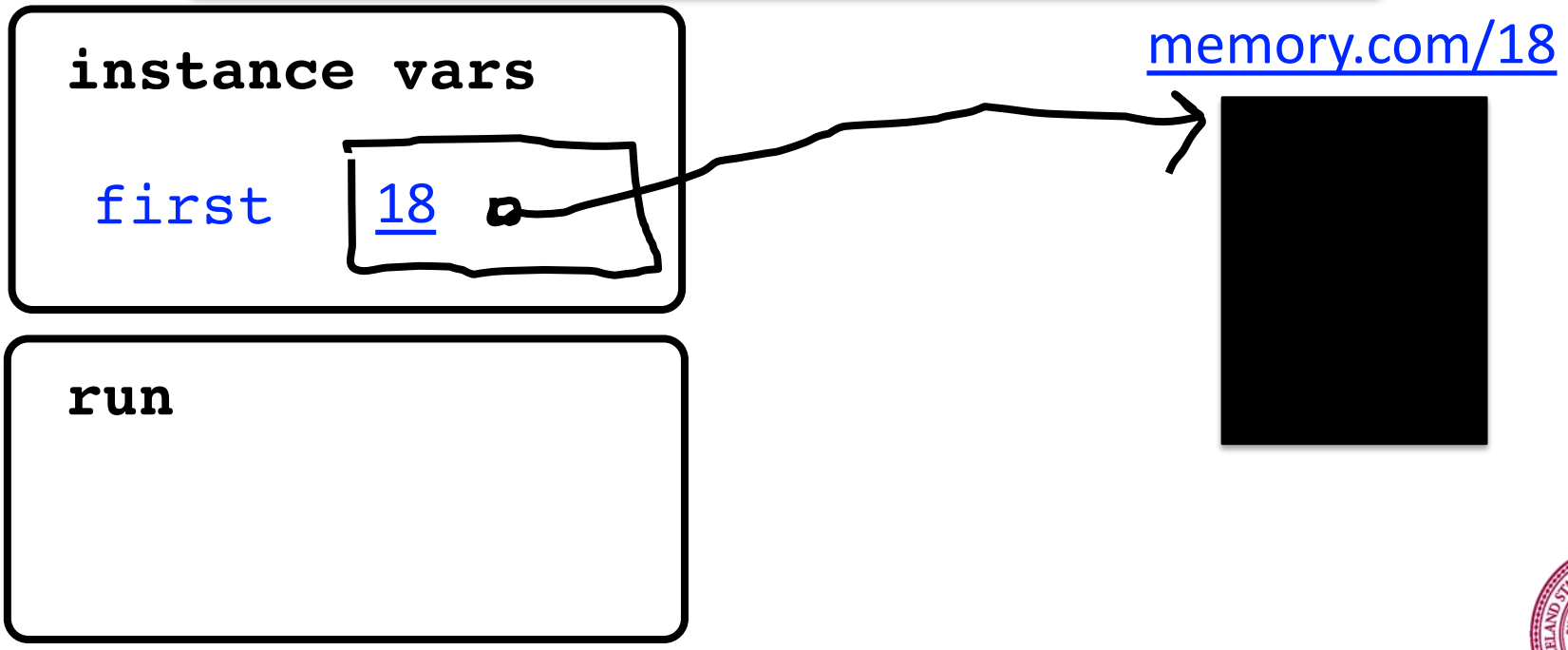
Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```



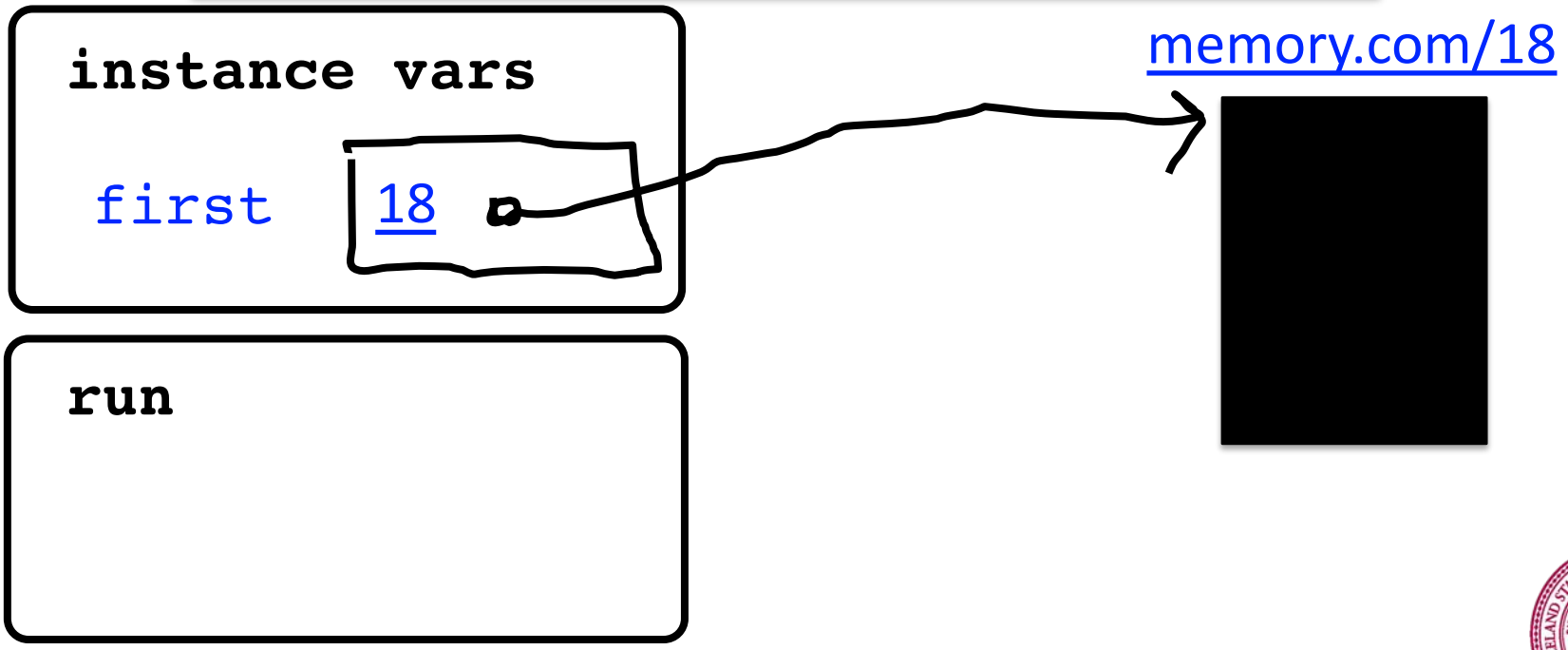
Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```



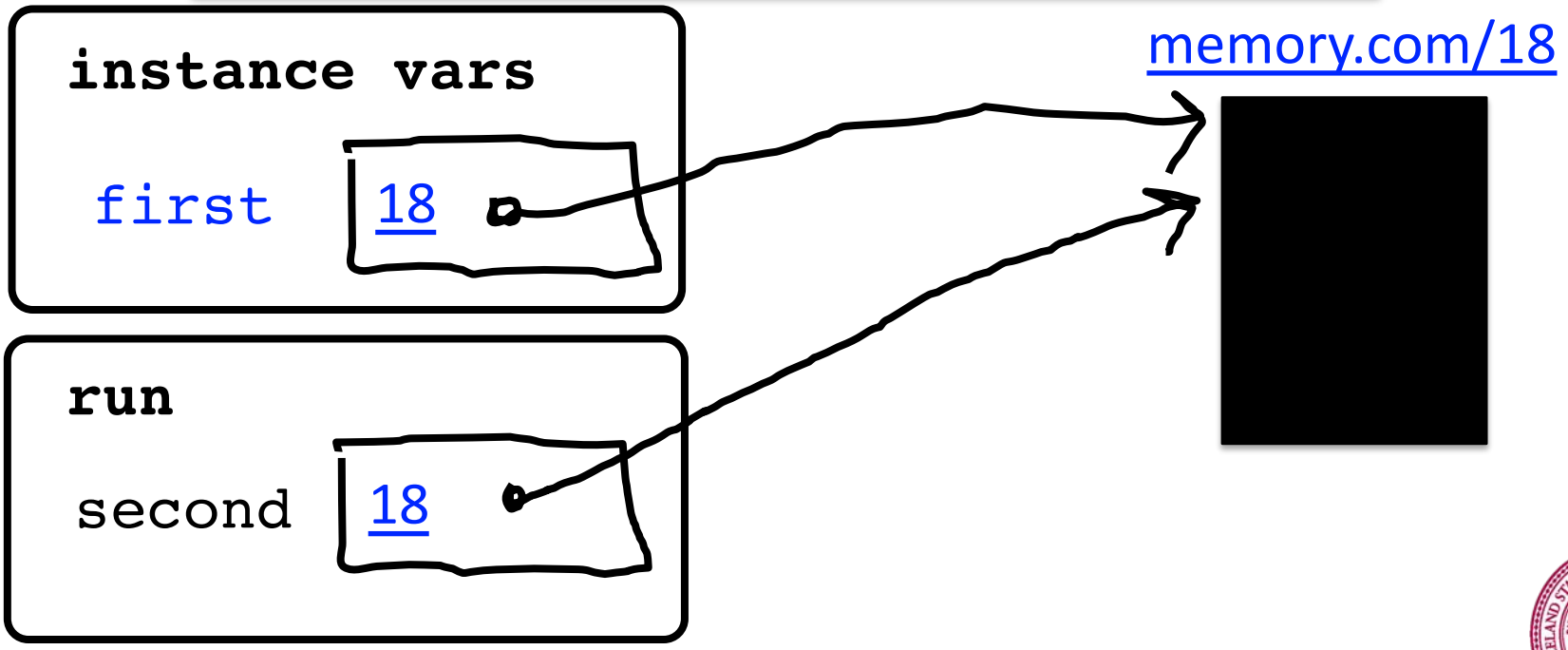
Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```



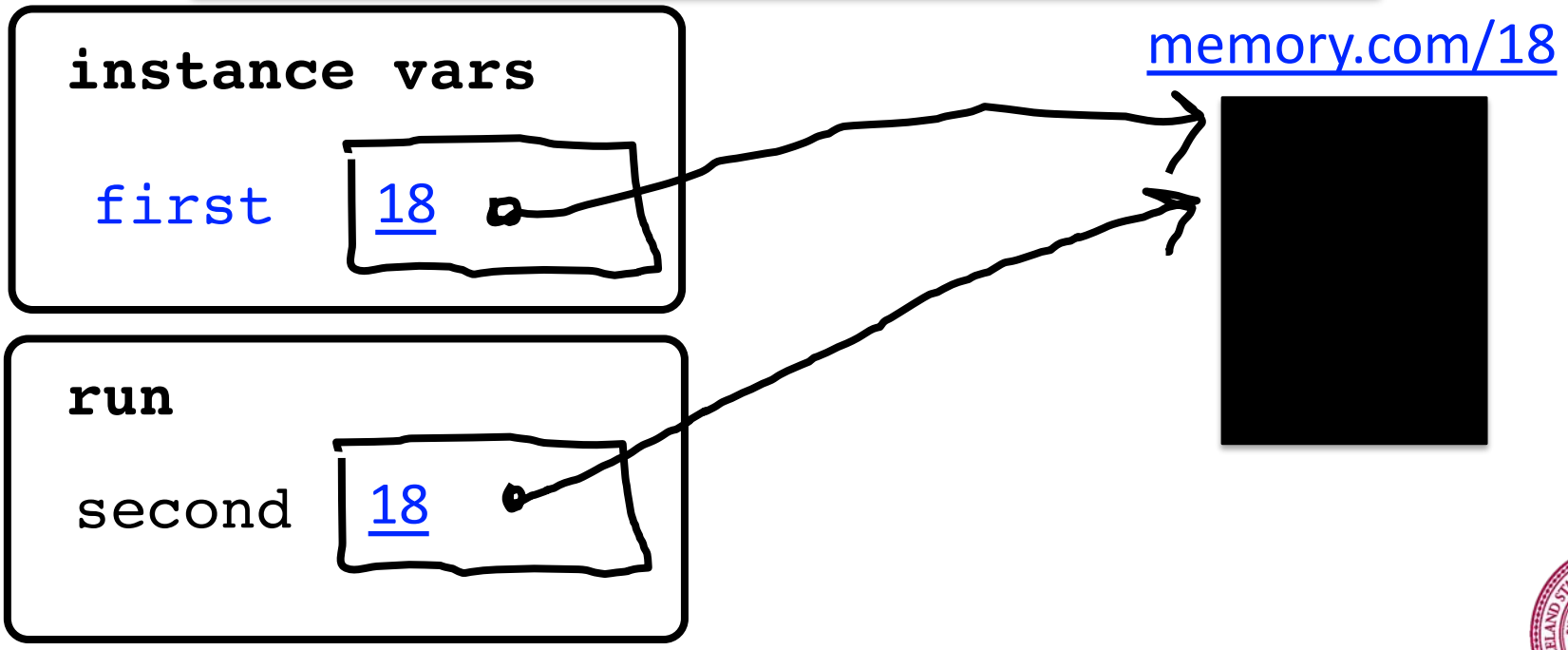
Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```



Who thinks this prints **`true`**?

```
private GRect first = new GRect(20, 30);
public void run() {
    first.setFilled(true);
    add(first, 0, 0);
    GObject second = getElementAt(1, 1);
    println(first == second);
}
```




You can do it!