

# CSE 11

# Accelerated Intro to Programming

## Lecture 9

Greg Miranda, Spring 2021

# Announcements

- Quiz 9 due Monday @ 8am
- PA3 due Wednesday @ 11:59pm
- Survey 3 due tonight @ 11:59pm
- PA0.5 Resubmission due tonight @ 11:59pm
  - Or see a tutor during lab hours and demo it

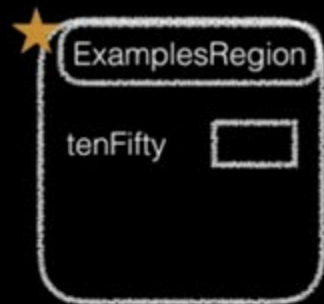
# Constructors

- Now that we understand the Stack, we have what we need to understand constructors

stack  
method stack frames

heap  
objects

```
class Point {  
    int x;  
    int y;  
    Point(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
  
}  
  
class ExamplesRegion {  
    Point tenFifty = new Point(10, 50);  
}
```



# Constructor Summary

- Constructors:



- Are special methods, called when **new** is used
- Are passed the newly-constructed object as this, and any arguments
- Typically assign values into fields using this.field = value

- When new is used:

- A fresh object, with a new reference is created with uninitialized fields
- The constructor with parameters that match the arguments is called
- The whole new expression evaluates to the new reference

```

class CircRegion2 {
    int radius;
    int cX;
    int cY;
    CircRegion2(int radius, int cX, int cY) {
        this.radius = radius;
        this.cX = cX + 100;
        this.cY = cY;
    }
}

class ExamplesRegion {
    CircRegion2 cr2 = new CircRegion(10, 50, 100);
}

```

Handwritten annotations for the constructor:  $10$ ,  $50$ ,  $100$  above the parameters;  $50 + 100 = 150$  next to the `this.cX` assignment.

A  
9

CircRegion	
radius	10
cX	150
cY	100

~~C~~ 9


CircRegion	
radius	10
cX	50
cY	100

B  
1

CircRegion	
radius	10
cX	50
cY	200

~~D~~ 1

CircRegion	
radius	110
cX	50
cY	100

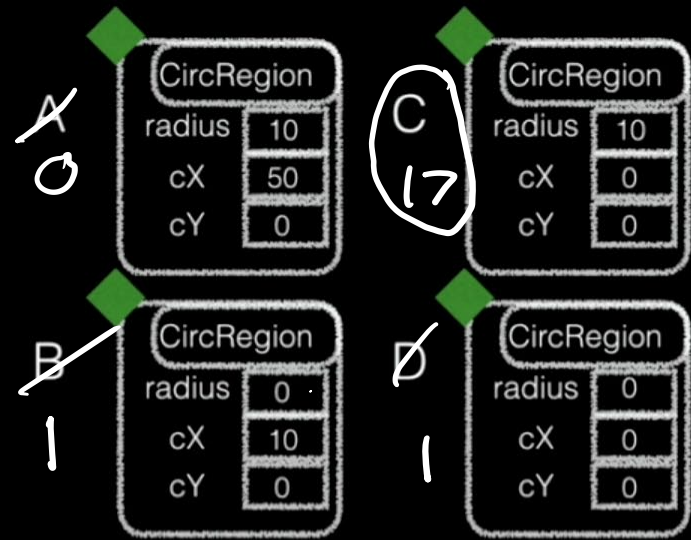
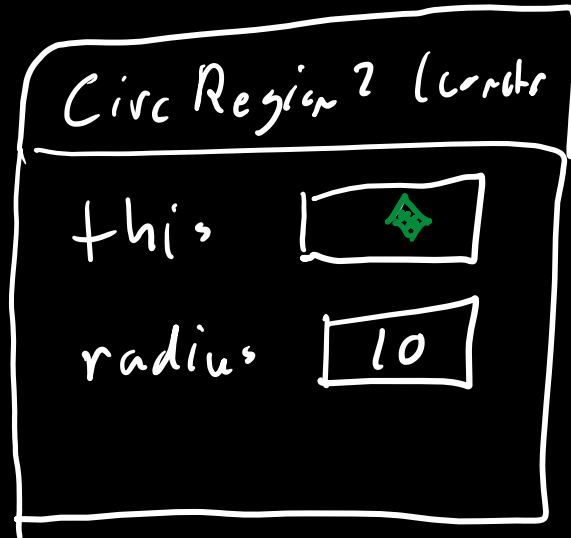
CircRegion (const)	
this	
radius	10
cX	50
cY	100

```

class CircRegion2 {
    int radius;
    int cX;
    int cY;
    CircRegion2(int radius) {
        this.radius = radius;
        this.cX = 0;
        this.cY = 0;
    }
}

class ExamplesRegion {
    CircRegion2 cr2 = new CircRegion(10);
}

```



# Tester

- import tester.\*;
  - tester.jar – java archive
    - Libraries that contain classes that we can use in our own code
      - Tester
  - Tester class allows us to create methods to unit test our code
    - Unit testing – compare actual values versus expected values
      - t.checkExpect(<actual value>, <expected value>);
    - Goal: get all tests to pass
      - Confidence that your code/solution is correct

→ *boolean checkExpect ( ? act , ? value );*



# Local Variables

- Local variables are defined inside the body of a method
  - They are 'local' to the method in which they are defined in
- Used temporarily while the method is running, then are removed
  - Similar to parameters
  - Added to the stack frame for the method
- No default value
  - Must be assigned a value before it's read from
    - i.e. used as an expression

