

Calling constructors from constructors

```

a0
equals(Object) toString() Object
x 20 y 30 Shape
Shape() Shape(int, int)
getX() getY() toString()

```

(x, y): position of upper-left corner of the bounding box of the Shape

CS101J, Cornell 1

Calling constructors from constructors

```

a0
equals(Object) toString() Object
x 20 y 30 Shape
Shape() Shape(int, int)
getX() getY() toString()

```

To call another constructor in the same class, use `this(...)` instead of `Shape(...)`

Principle: Use previously written methods as much as possible, in order to save time and effort in writing a program and also in testing and debugging it.

CS101J, Cornell 1

The constructor call must be the first statement in the constructor

```

a0
equals(Object) toString() Object
x 20 y 30 Shape
Shape() Shape(int, int)
getX() getY() toString()

```

Rule: A constructor call `this(...)` within a constructor must be the first statement in the constructor body.

```

public Shape() {
  x = 20;
  this(100, 100);
}

```

DOESN'T WORK!

CS101J, Cornell 2

Calling a superclass constructor

```

a0
equals(Object) toString() Object
x 20 y 30 Shape
Shape() Shape(int, int)
getX() getY() toString()
radius 10 Circle
Circle(int, int, int) area()
getRadius() toString()

```

Rule: Make the first statement of the constructor a call `super(...)` so that inherited fields are initialized first. This superclass constructor call cannot be placed elsewhere.

CS101J, Cornell 3