

CSE 11

Accelerated Intro to Programming

Lecture 10

Greg Miranda, Spring 2021

Announcements

- Quiz 11 due Friday @ 8am
- PA3 due tonight @ 11:59pm
- Survey 4 due Friday @ 11:59pm
- Week 6, 9, and finals week for take-home exams

```

        interface Region {
            boolean contains(Point p);
        }

class CircleRegion
    implements Region {
    ...
    public boolean contains(Point p)
    { ... }
}

class SquareRegion
    implements Region {
    ...
    public boolean contains(Point p)
    { ... }
}

```

What is the value of
the b1 field?

```

class UnionRegion {
    Region r1, r2;
    UnionRegion(Region r1, Region r2) { ... }
    public boolean contains(Point p) {
        return this.r1.contains(p) ||
               this.r2.contains(p);
    }
}

class ExamplesRegion {
    Region circ = new CircleRegion(new Point(10, 5), 4);
    Region square = new SquareRegion(new Point(5, 6), 8);
    UnionRegion ur = new UnionRegion(this.square, this.circ);
    boolean b1 = this.ur.contains(new Point(13, 5));
}

```

20 (A) true | C: error
4 B: false

Using Interfaces

- Add Region interface
- Update RectRegion & CircRegion

Intersect Region

```
interface Region {  
    boolean contains(Point p);  
}
```

```
class IntersectRegion implements Region {  
    Region r1;  
    Region r2;  
    IntersectRegion(Region r1, Region r2) {  
        this.r1 = r1;  
        this.r2 = r2;  
    }  
    public boolean contains(Point p) {  
        return this.r1.contains(p) && this.r2.contains(p);  
    }  
}
```

```

class IntersectRegion implements Region {
    ...
    public boolean contains(Point p) {
        return this.r1.contains(p) && this.r2.contains(p);
    }
}

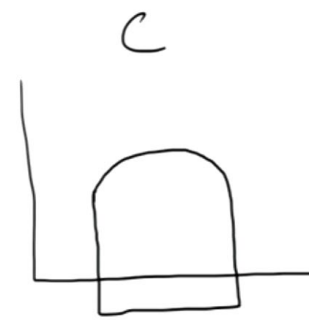
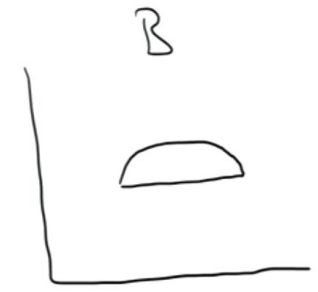
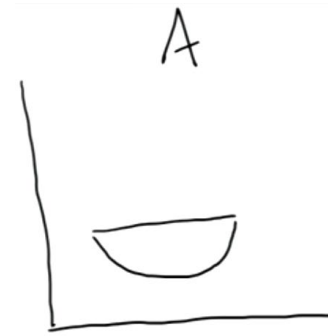
```

```

class ExamplesRegion {
    Region circ1 = new CircleRegion(new Point(10, 5), 4.0);
    Region sq = new SquareRegion(new Point(10, 1), 8.);
    Region ir = new IntersectRegion(this.circ1, this.sq);

    //What region is represented in toDraw?
    Region toDraw = ir;
}

```



SubtractRegion

- Write a new class called SubtractRegion
 - That also implements Region
- Represents all the points in region1 that aren't in region2
 - Start with a shape and subtract another shape from it
 - Subtract a circle from another circle to create a ring
- Write the class
 - Fields and constructor
 - What does the contains method look like?

Something more complicated

- What if we wanted a region that was 3 circles next to each other:
- How could we construct an example of this?

Inheritance

- New terms
 - abstract class
 - extends