CSE 11
Accelerated Intro to Programming
Lecture 10

Greg Miranda, Spring 2021

This lecture is being recorded

Announcements

- Quiz 10 due Wednesday @ 8am

- PA3 due Wednesday @ 11:59pm

- Survey 4 due tonight @ 11:59pm

WeeklyPay

- weeklyPay: takes a number of hours worked and an hourly rate, and returns the pay with overtime (over 40 hours) counting as double the rate

```
class CircleRegion {              class SquareRegion {
  …                                 …
  boolean contains(Point p)         boolean contains(Point p)
  { … }                             { … }
}                                 }
```

Both classes have a method with the **same header**

We can write an **interface** with the shared method

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

```
class CircleRegion
  implements Region {

  …
  public boolean contains(Point p)
  { … }
}
```

```
class SquareRegion
  implements Region {

  …
  public boolean contains(Point p)
  { … }
}
```

Both classes have a method with the **same header**

We can write an **interface** with the shared method

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

```
class CircleRegion              class SquareRegion
  implements Region {             implements Region {

  …                               …
  public boolean contains(Point p)  public boolean contains(Point p)
  { … }                           { … }
}                               }
```

Both classes have a method with the **same header**

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

```
class CircleRegion                     class SquareRegion
   implements Region {                    implements Region {
  …                                      …
   public boolean contains(Point p)       public boolean contains(Point p)
   { … }                                  { … }
}                                      }
```

```
class ExamplesRegion {
   Region circ = new CircleRegion(new Point(10, 5), 4));
   Region square = new SquareRegion(new Point(5, 6), 8));
}
```

We can use the same
interface type
for references of classes that implement it

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

```
class CircleRegion                          class SquareRegion
  implements Region {                         implements Region {
  Point center;                               Point center;
  int radius;                                 int sideLength;
  public boolean contains(Point p)            public boolean contains(Point p)
  { … }                                       { … }
}                                           }
```

```
        class ExamplesRegion {
            Region circ = new CircleRegion(new Point(10, 5), 4));
            Region square = new SquareRegion(new Point(5, 6), 8));
```
double ~~int~~ `num = circ.radius;`
```
            }
```

What is stored in the num field?

A: 10   C: 5

B: 8   D: it's an error

E: something else

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

```
class CircleRegion                          class SquareRegion
    implements Region {                         implements Region {
  Point center;                             Point center;
  int radius                                int sideLength;
  public boolean contains(Point p)          public boolean contains(Point p)
  { … }                                     { … }
}                                           }
```

```
class ExamplesRegion {
    Region circ = new CircleRegion(new Point(10, 5), 4);
    Region square = new SquareRegion(new Point(5, 6), 8);
    boolean contains1 = circ.contains(new Point(7, 6));
}
```

What is stored in
the contains1 field?

A: true   C: error

B: false

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

class CircleRegion                  class SquareRegion
  implements Region {                 implements Region {

  …                                   …
  public boolean contains(Point p)    public boolean contains(Point p)
  { … }                               { … }
}                                   }



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

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

class CircleRegion                    class SquareRegion
    implements Region {                   implements Region {

    …                                     …
    public boolean contains(Point p)      public boolean contains(Point p)
    { … }                                 { … }
}                                     }


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));
}
```

What is the value of the b1 field?

A: true   C: error

B: false