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

```
class SquareRegion {
  …
  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**

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

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

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

```
class SquareRegion
  implements Region {
  Point center;
  int sideLength;
  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 num = circ.radius;
}
```

What is stored in the num field?

A: 10 C: 5

B: 8 D: it's an error

Using an interface type, we can only use methods on the interface.

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

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

```
class SquareRegion
  implements Region {
  Point center;
  int sideLength;
  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

B: false

C: error

```
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));
}
```

A: true

B: false

C: error