



[Course](#) [Progress](#)

[Course](#) > [Readings](#) > [Reading 2: Recursive Data Types](#) > [Questions](#)

[< Previous](#)



























[Next >](#)

Questions

 [Bookmark this page](#)

isEmpty? No, isImplemented!

3/3 points (graded)

```
isEmpty : IList -> boolean
isEmpty(Empty) = true
isEmpty(Cons(first:E, rest:IList)) = false
```

Let's implement `IList.isEmpty`.

```
public interface IList<E>
// ...
/**
 * @return true iff this list is empty
 */
```

public boolean isEmpty

✓ Answer: public boolean isEmpty(); or boolean isEmpty();

Explanation

`boolean isEmpty();` is also correct, but note that all interface methods are automatically public.

```
}
```

...

```
class Empty<E> implements IList<E> {
// ...

    public boolean isEmpty() {
```

return true;

✓ Answer: return true;

```
    }
}
```

...

```
class Cons<E> implements IList<E> {
    private E e;
    private IList<E> rest;
// ...


    public boolean isEmpty() {
```

return false;

✓ Answer: return false;

```
    }
}
```

Submit

 Show Answer

 Answers are displayed within the problem

[< Previous](#) [Next >](#)

[Open Learning Library](#)

[About](#)

[Accessibility](#)

[All Courses](#)

[Why Support MIT Open Learning?](#)

[Help](#)

[Connect](#)

[Contact](#)

[Twitter](#)

[Facebook](#)