Part 6 | Final Exam | 6.005.1x Courseware

1. Course , current location
2. Progress

Part 6

Implementing an Interface

0.0/3.0 points (graded)

Here is an extremely simplified Set interface with only one operation, and one simple implementation class for it:

```
/** Represents an immutable set of elements of type E. */
interface Set<E> {
    /** @return true iff this set contains e as a member */
    public boolean contains(E e);
}
/** A Set<E> that contains every E. */
class Universe<E> {
    /** Make a universe. */
    public Universe() { }
    /** @return always true since this set contains every e */
    public boolean contains(E e) { return true; }
}
```

`Universe`**doesn't** correctly implement the `Set` interface because (choose all good answers):

unanswered

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Interfaces and Implementations

0.0/3.0 points (graded)

Assume the following lines of code are run in sequence, and that any lines of code that don't compile are simply commented out so that the rest of the code can compile.

The code uses two methods from java.util.Collections, so you might need to consult the documentation.

Choose the **most specific answer** to each question.

```
Set<String> set = new HashSet<String>();
```

The `set` variable now points to:

unanswered

```
set = Collections.unmodifiableSet(set);
```

The `set` variable now points to:

unanswered

```
set = Collections.singleton("glorp");
```

The `set` variable now points to:

unanswered

```
set = new Set<String>();
```

The `set` variable now points to:

unanswered

```
List<String> list = set;
```

The `set` variable now points to:

unanswered

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Tolerance

0.0/3.0 points (graded)

Knowing that floating-point calculations can have some error, suppose we try to implement Double.equals() with tolerance:

```
  class Double {
    private final double value;
    @Override public boolean equals (Object thatObject) {
        if (!(thatObject instanceof Double)) return false;
        Double that = (Double) thatObject;
        return that.value - this.value < 0.01;
    }
}
```

Which properties of an equivalence relation are violated by this equals() method?

unanswered

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Equality

0.0/2.0 points (graded)

Suppose you want to show that an equality operation is buggy because it isn't symmetric. How many objects do you need for a counterexample to symmetry?

unanswered

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Object Contract

0.0/3.0 points (graded)

If a type is correctly obeying the Object contract, which of the following are true?

unanswered

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.