Part 1 | Final Exam | 6.005.1x Courseware

1. Course , current location
2. Progress

Part 1

Checking 1

0.0/1.0 point (graded)

This Java code has a bug. Is it caught statically, dynamically, or not at all?

```
double oneThird = 1/3;
```

unanswered

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

Checking 2

0.0/1.0 point (graded)

This Java code has a bug. Is it caught statically, dynamically, or not at all?

```
int sum = 0;
int n = 0;
int average = sum/n;
```

unanswered

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

Assumptions

0.0/3.0 points (graded)

In Java, which of the following assumptions could be documented by type declarations and statically checked by the Java compiler?

unanswered

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

Java Types

0.0/2.0 points (graded)

Which of the following are legal Java types?

unanswered

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

Part 2 | Final Exam | 6.005.1x Courseware

1. Course , current location
2. Progress

Part 2

DRY

0.0/3.0 points (graded)

Which of the following are true about Don't Repeat Yourself (DRY)?

unanswered

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

Reading Java

0.0/2.0 points (graded)

```
public static boolean leap(int y) {
    String tmp = String.valueOf(y);
    if (tmp.charAt(2) == '1' || tmp.charAt(2) == '3' || tmp.charAt(2) == 5
            || tmp.charAt(2) == '7' || tmp.charAt(2) == '9') {
        if (tmp.charAt(3)=='2'||tmp.charAt(3)=='6') return true; /*R1*/
        else
            return false; /*R2*/
    }else{
        if (tmp.charAt(2) == '0' && tmp.charAt(3) == '0') {
            return false; /*R3*/
        }
        if (tmp.charAt(3)=='0'||tmp.charAt(3)=='4'||tmp.charAt(3)=='8')return
true; /*R4*/
    }
    return false; /*R5*/
}
```

What happens when you call:

```
leap(2015)
```

unanswered

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

Partitioning

0.0/2.0 points (graded)

Consider the following specification:

```
/**
 * Reverses the end of a string.
 *
 *                           012345                     012345
 * For example: reverseEnd("Hello, world", 5) returns "Hellodlrow ,"
 *                                <----->                   <----->
 *
 * With start == 0, reverses the entire text.
 * With start == text.length(), reverses nothing.
 *
 * @param text    non-null String that will have its end reversed
 * @param start   the index at which the remainder of the input is reversed,
 *                requires 0 <= start <= text.length()
 * @return input text with the substring from start to the end of the string
reversed
 */
public static String reverseEnd(String text, int start)
```

Which of the following are reasonable partitions for the start parameter? Check all that apply.

unanswered

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

sin(x)

0.0/4.0 points (graded)

Which of the following are reasonable partitions for `sin(x)`?

unanswered

Which of the following are boundary values for `sin(x)`?

unanswered

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

Test-First Programming

0.0/3.0 points (graded)

Which of these techniques are useful for choosing test cases in test-first programming, before any code is written? Check all that apply.

unanswered

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

Regression Testing

0.0/2.0 points (graded)

A regression test case:

unanswered

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

Part 3 | Final Exam | 6.005.1x Courseware

1. Course , current location
2. Progress

Part 3

Preconditions

0.0/4.0 points (graded)

Alice writes the following code:

```
public static int gcd(int a, int b) {
    if (a > b) {
        return gcd(a-b, b);
    } else if (b > a) {
        return gcd(a, b-a);
    }
    return a;
}
```

Bob writes the following test:

```
@Test public void gcdTest() {
    assertEquals(6, gcd(24, 54));
}
```

The test passes!

Alice should write `a > 0` in the precondition comment of `gcd`

unanswered

Alice should write `b > 0` in the precondition comment of `gcd`

unanswered

Alice should write `gcd(a, b) > 0` in the precondition comment of `gcd`

unanswered

Alice should write `a and b are integers` in the precondition comment of `gcd`

unanswered

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

Valid Test Cases

0.0/3.0 points (graded)

Given this specification:

```
static int find(int[] arr, int val)
  requires: arr[0] == val
  effects:  returns index i such that arr[i] == val
```

Which are valid test cases for `find`?

unanswered

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

Contracts

0.0/3.0 points (graded)

Consider this spec:

```
static int find(int[] arr, int val)
  requires: val occurs exactly once in arr
  effects:  returns index i such that arr[i] = val
```

As the implementer of `find`, which are legal? Check all that apply.

unanswered

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

Spec Strength

0.0/4.0 points (graded)

Assuming everybody follows the spec, what can an implementor do, without looking at or changing any clients?

unanswered

Assuming everybody follows the spec, what can a single client do, without looking at or changing the implementation or any other clients?

unanswered

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

Stronger and Weaker

0.0/3.0 points (graded)

Which of the following can be true about a pair of specifications *A* and *B*? Check all that apply.

unanswered

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

Part 4 | Final Exam | 6.005.1x Courseware

Part 4

An Interview

0.0/4.0 points (graded)

Suppose you are listening to an interview candidate for a software development job. Which statements show that they know what they're talking about?

unanswered

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

final

0.0/3.0 points (graded)

Consider the following code, executed in order:

```
char vowel0 = 'a';
final char vowel1 = vowel0;

String vowel2 = vowel1 + "eiou";
final String vowel3 = vowel2;

char[] vowel4 = new char[] { vowel0, 'e', 'i', 'o', 'u' };
final char[] vowel5 = vowel4;
```

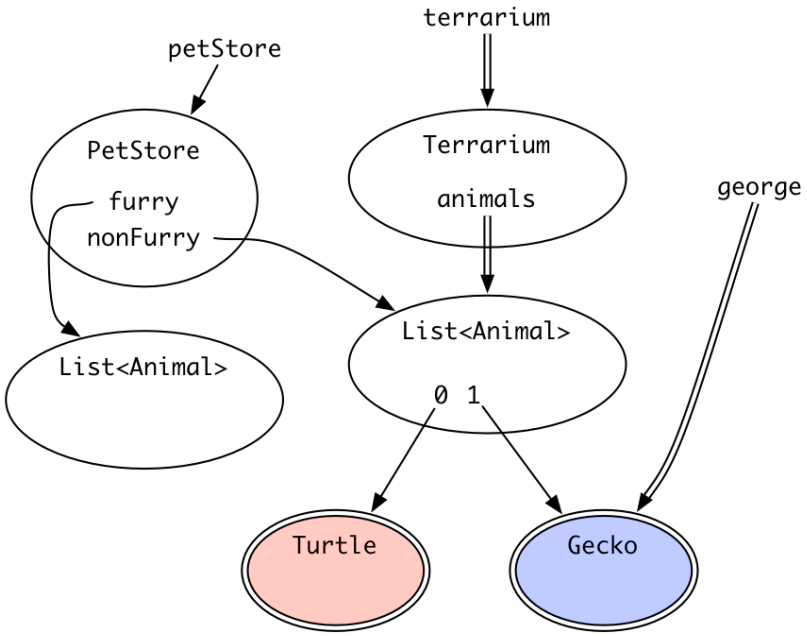
Which of the following statements are legal Java (i.e. produce no compiler error if placed after the code above)? Check all that apply.

unanswered

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

Mutability

0.0/4.0 points (graded)



Is it possible that a client with the variable `terrarium` could modify the `Turtle` object in red?

unanswered

Is it possible that a client with the variable `petStore` could do something such that a client with the variable `terrarium` could no longer access the `Gecko` in blue?

unanswered

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

Reassignment and Mutation

0.0/3.0 points (graded)

Each of the following lines of Java code declares a variable.

```
String a = "hello";
static String b = "hello";
final String c = a;
final String d = b;
```

For which of the variables does Java guarantee that it won't be reassigned?

unanswered

Each of the following lines of Java code creates a new object.

```
String x = "hello";
final List<String> y = new ArrayList<>();
final List<String> z = Collections.unmodifiableList(y);
```

For which of the objects (denoted by the variable pointing to it) does Java guarantee that the object won't be mutated?

unanswered

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

Immutability

0.0/3.0 points (graded)

Which of the following are correct? Check all that apply.

unanswered

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

Part 5 | Final Exam | 6.005.1x Courseware

1. Course , current location
2. Progress

Part 5

Representations

0.0/2.0 points (graded)

Consider an abstract data type `Bool`. The type has the following operations:

```
true : void → Bool
false : void → Bool
and : Bool × Bool → Bool
or : Bool × Bool → Bool
not : Bool → Bool
```

...where the first two operations construct the two values of the type, and last three operations have the usual meanings of logical *and*, logical *or*, and logical *not* on those values.

Which of the following are possible ways that `Bool` might be implemented, and still be able to satisfy the specs of the operations? Check all that apply.

unanswered

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

Rep Invariants

0.0/3.0 points (graded)

```
/** An immutable rational number. */
class RatNum {
    private int a, b;
    ...
}
```

Which of the following are plausible rep invariants for RatNum?

unanswered

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

Abstract Data Types

0.0/4.0 points (graded)

Which of the following should be known (visible and documented) to the client of an abstract data type? Check all that apply.

unanswered

Which of the following should be known (visible and documented) to the maintainer of an abstract data type? Check all that apply.

unanswered

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

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.