EECS1022 Programming for Mobile Computing (Winter 2021)

Q&A - Lectures W3

Monday, February 1

Example Solutions

↳ vs. your solution.

Prog. Test 1

→ Lab1
Lab2

- Utilities
- TestUtilities

mandatory

→ implement methods

→ some tests

↳ more tests to grade submission

(⇒ you should try more input value on tests)

a. more junit tests.

b. App.java

DeMorgan Law of Conjunction: Example (3)

```
if (i < 0 && i > 10) { /* Action 1 */ }
else { /* Action 2 */ }
```

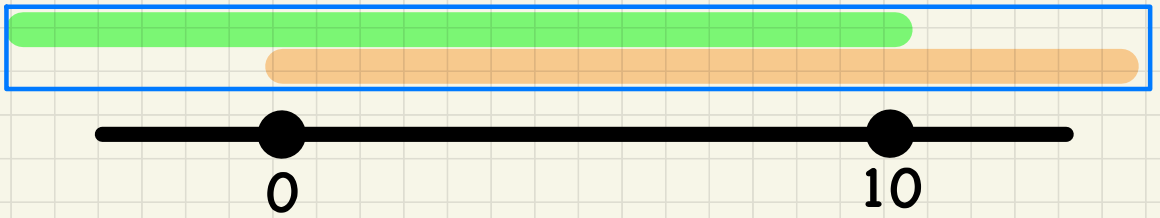
- **When** is *Action 1* executed? *false*
- **When** is *Action 2* executed? *true* (i.e., `i >= 0 || i <= 10`)

$\equiv !(i<0 \;\&\&\; i>10)$

$\equiv !(i<0) \;||\; !(i>10) \equiv i \geq 0 \;||\; i \leq 10$

|||

True



De Morgan

① $!(a \;\&\&\; b) \equiv !a \;||\; !b$

② $!a \;\&\&\; !b \equiv !(a \;||\; b)$

/*
*
*/

3~4.

```
public static String getArea(double r){
    String result = "";
    if( r < 0){
        result = "Error--";
    }
    return result;
}
```

"..." A wins
B wins
tie

1. Read the comments for method (problem).
2. Go JUnit tests (5 ~ <10).
   test_01.

failure trace
↳ expected vs. actual
- scan code

Given: 5 tests.
Addition: 4 tests.

3 tests

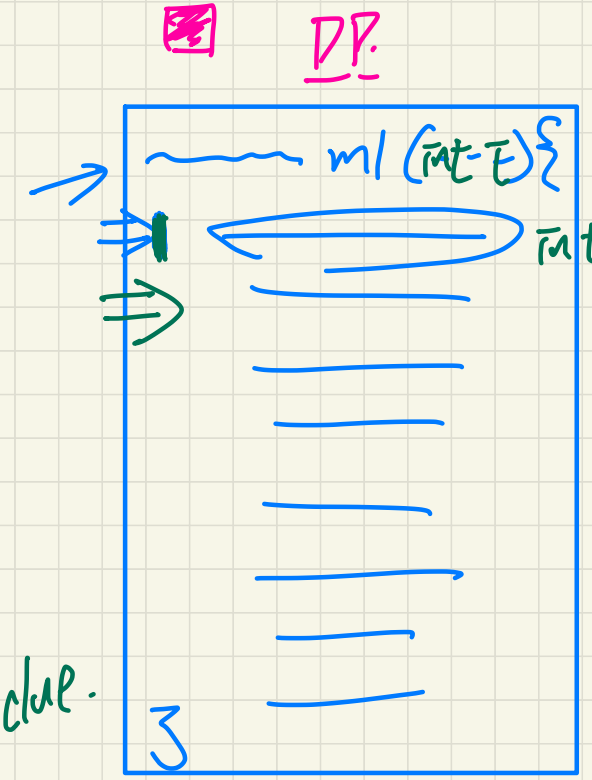
$\frac{3}{5+4}$.

App.java ✓

Structure of code

↳ ✗

↳ marks?



Exp.

t*t+1

Debugger

⇒ use it when you have no clue.

boolean isPositive(int i)

U.isPositive(2) → T

-4

0