CLASS 4 - THE ART OF CLEAN CODE - FUNDAMENTALS - CONTINUED

venks@iiit.ac.in

January 13, 2020

IIIT Hyderabad

CONTENTS

CONTENTS

- Recap of last class

CONTENTS

- Recap of last class
- Introduction

CONTENTS

- Recap of last class
- Introduction
- Functions

CONTENTS

- Recap of last class
- Introduction
- Functions
- Comments

CONTENTS

- Recap of last class
- Introduction
- Functions
- Comments
- Clean Tests

CONTENTS

- Recap of last class
- Introduction
- Functions
- Comments
- Clean Tests
- Error Handling

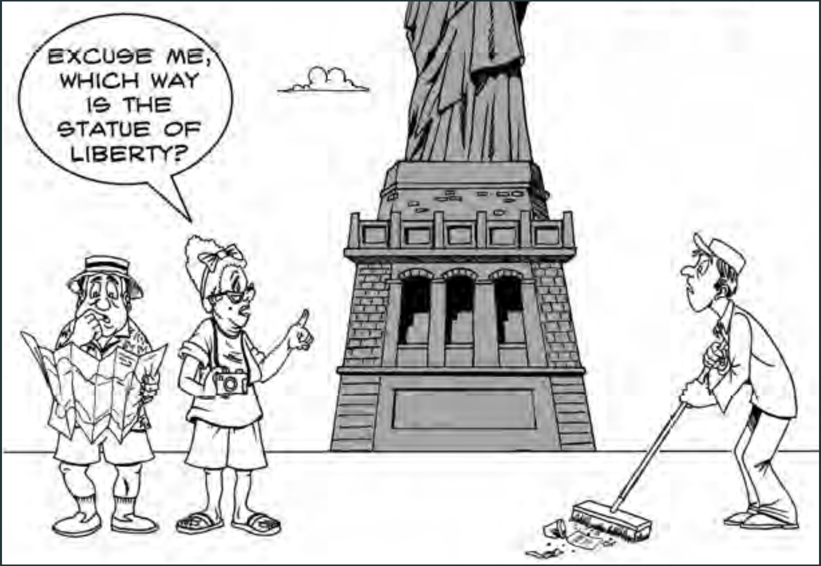
RECAP OF LAST CLASS

COMMENTS

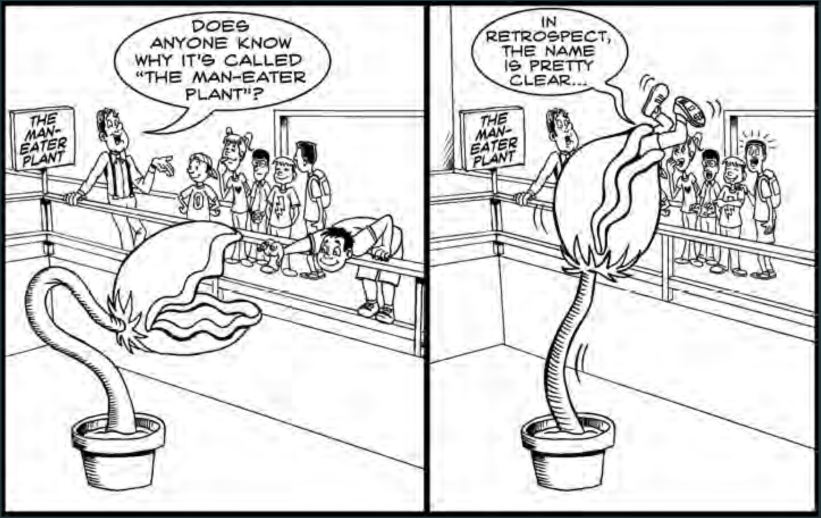
COMMENTS

”Don’t comment bad code-Rewrite it.”

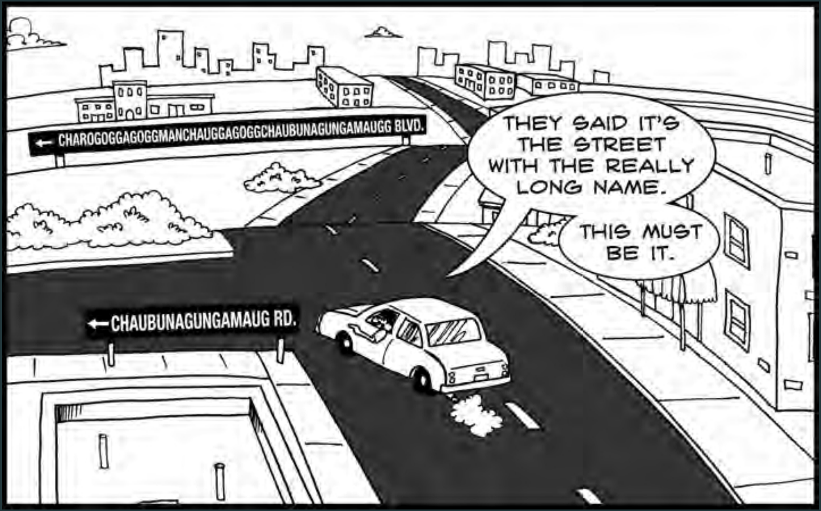
WHAT NOT TO COMMENT



COMMENTS


THE MAN-EATER PLANT
DOES ANYONE KNOW WHY IT'S CALLED "THE MAN-EATER PLANT"?
THE MAN-EATER PLANT
IN RETROSPECT, THE NAME IS PRETTY CLEAR...

RIGHT LENGTH


← CHAROGOGGAGOGGMANCHAUGGAGOGGCHAUBUNAGUNGAMAUGG BLVD.
← CHAUBUNAGUNGAMAUG RD.
THEY SAID IT'S THE STREET WITH THE REALLY LONG NAME.
THIS MUST BE IT.

COMMENTS

”Prefer to explain yourself in Code.”

EXPLAIN YOURSELF IN CODE

```
// Check to see if the employee is eligible for full benefits
if ((employee.flags & HOURLY_FLAG) && (employee.age > 65))
```

Or

```
if (employee.isEligibleForFullBenefits())
```

BIG PICTURE COMMENTS

”One of the hardest things for a new team member to understand is the “big picture”—how classes interact, how data flows through the whole system, and where the entry points are.”

COMMENTS

1. README for the service for ”Big picture comment”. For example, this service uses this Implementation Pattern.
2. Write Javadocs for Public APIs.
3. Javadocs must follow the standard convention.
4. Javadocs for private methods are more of a distraction.

CLEAN TESTS

CLEAN TESTS

”Test code should be readable so that other coders are comfortable changing or adding tests.”

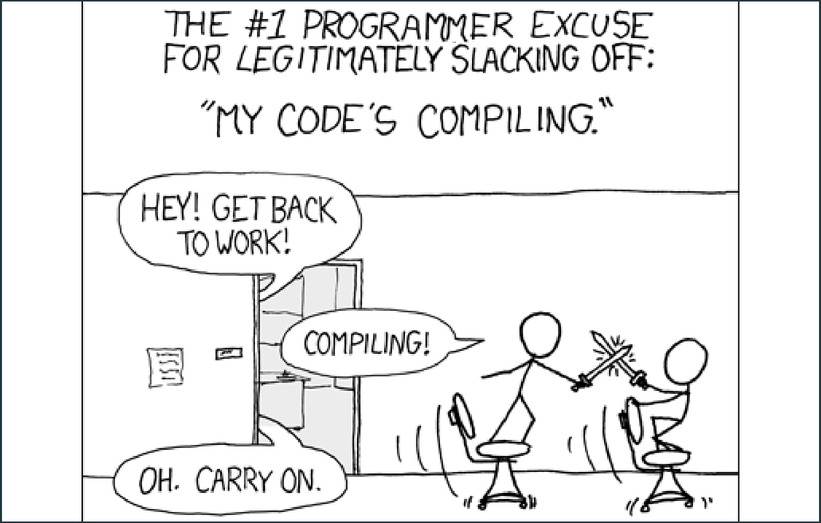
A SIMPLE TEST

```
import static org.junit.jupiter.api.Assertions.assertEquals;

import example.util.Calculator;

import org.junit.jupiter.api.Test;

class MyFirstJUnitJupiterTests {

    private final Calculator calculator = new Calculator();

    @Test
    void addition() {
        assertEquals(2, calculator.add(1, 1));
    }

}
```

CLEAN TESTS

1. Readability is the top priority.
2. Clarity, simplicity and density of expression.
3. Prioritize readability over performance.
4. One assert per test.
5. Single concept per test.

COMMENTS


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.

F.I.R.S.T.

1. Fast - Tests should be fast.
2. Independent - Tests should not depend on each other.
3. Repeatable - Tests should be repeatable in any environment, Prod, Devo, etc.
4. Self-validating - Should have a boolean output.
5. Timely - Write them along with production code or just before.

ERROR HANDLING

ERROR HANDLING

”Error handling is important, but if it obscures logic, it’s wrong.”

ERROR HANDLING

1. Prefer exceptions to error codes.
2. Always write try-catch-finally statement first.
3. The battle is over. Use unchecked exceptions.
4. Do not violate Open/Closed Principle.

RESOURCES

1. Clean Code by Robert C Martin

Questions?