CLASS 3 - THE ART OF CLEAN CODE - FUNDAMENTALS

venks@iiit.ac.in

January 9, 2020

IIIT Hyderabad

CONTENTS

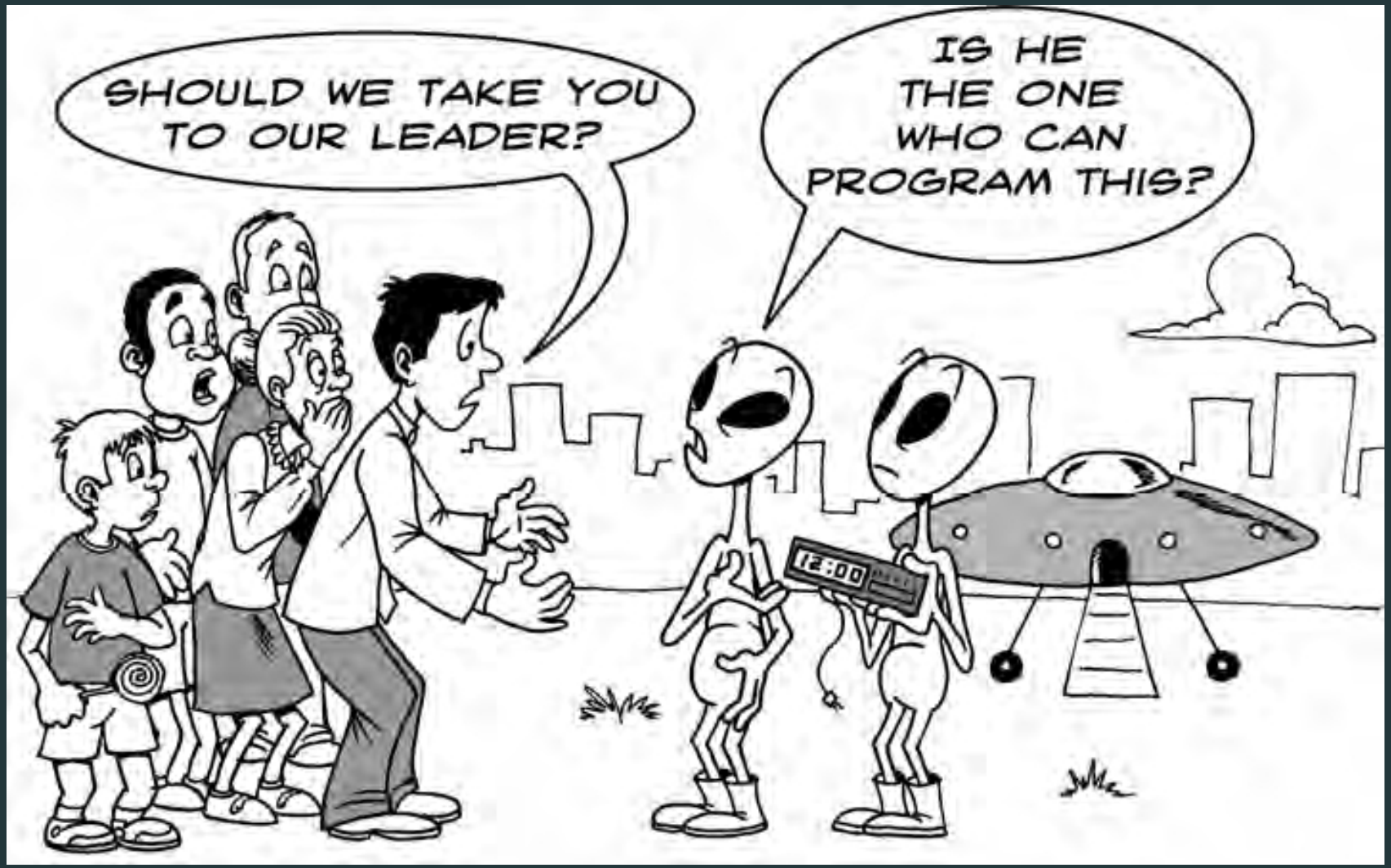
CONTENTS

CONTENTS

RECAP OF LAST CLASS

INTRODUCTION

INTRODUCTION



BROKEN WINDOW METAPHOR

”A building with broken windows looks like nobody cares about it. So other people stop caring. They allow more windows to become broken. Eventually they actively break them. One broken window starts the process toward decay.”

WHAT IS CLEAN CODE?

1. Clean code is simple and direct.
2. Clean code reads like well-written prose.
3. Clean code never obscures the designer's intent but rather is full of crisp abstractions and straightforward lines of control.
4. Clean code does one thing.
5. Clean code always looks like it was written by someone who cares.

TERMINOLOGY

1. Always.
2. Never.
3. Usually applicable.
4. Usually not applicable.

FUNCTIONS

FUNCTIONS

FUNCTIONS

”As small as possible, as large as necessary.”

FUNCTIONS

1. Should be small.
2. Must do exactly one thing.
3. Write lots of them, ensure they are small.
4. One level of abstraction per function.

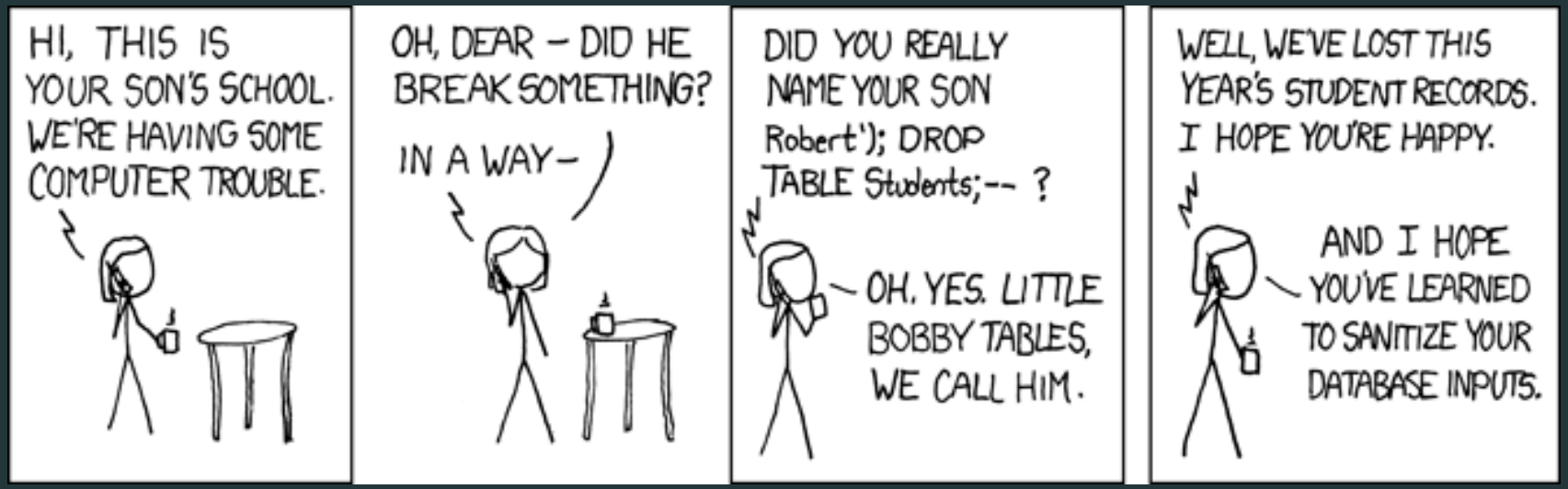
METHOD ARGUMENTS

”The ideal number of arguments for a function is zero. Next comes one, followed closely by two. Three arguments are best avoided. More than three requires specific justification and should never be used.”

ARGUMENT OBJECTS

```
Circle makeCircle(double x, double y, double radius);
Circle makeCircle(Point center, double radius);
```

VALIDATE INPUTS



METHODS

1. Validate your inputs.
2. Use verb/noun naming. (writeField, assertExpectedEqualsActual)
3. Avoid flag arguments.
4. Avoid passing and returning nulls.

OPTIONAL

```
Optional<Integer> possible = Optional.of(5);
possible.isPresent(); // returns true
possible.get(); // returns 5

Optional<Integer> notPossible =  Optional.absent();
notPossible.isPresent(); // returns false

Optional<Integer> getEmployeeId() {
    if(existsInDatabase()) {
        return Optional.of(valueInDatabase());
    }

    return Optional.absent();
}
```

SIDE EFFECTS AND COMMAND QUERY SEPARATION

Avoid side effects.

Either do something or answer something. Not both.

```
if(updateSomething(newValue))

if(existsSomething(value) {
   updateSomething(newValue)
}
```