# Obfuscation: Week Two

Caleb Parten and Carlo Velarde

### Overview

- Testing different obfuscation methods
- Analysis
- New prompting

#### **Testing Obfuscation Methods**

- Dead code obfuscation
- Naming obfuscation
- What did we test for?
  - Equality
  - Correct implementation

let name = "carlo"
console.log(name)

obfus cation Fail

let 01\_2zd = "carlo" console.log(name)

Success

function foo(name){
return name + "
Velarde"};

let name = "Carlo" foo(name)

console.log(name

let name = "Carlo"
console.log(name)

obfusc ation

#### **Testing Parameters**

- Equality
  - Should compile and output
  - Output must be direct match
- Correct implementation
  - Must use specified obfuscation method
  - Mixing obfuscation methods == False
  - Output can be off
- Successful obfuscation
  - True for equality AND True for implementation

## **Completed Table**

obfuscate output	obfuscation type	equal	implemented obfuscation
Area of circle: 78.53981633974483	Dead Code Obfuscation	FALSE	TRUE
10 kilometers is 6.21371 miles	Dead Code Obfuscation	TRUE	TRUE
Error: Command '['node', '-e', 'const stri	Dead Code Obfuscation	FALSE	TRUE
Error: Command '['node', '-e', 'const str	Dead Code Obfuscation	FALSE	TRUE
Error: Command '['node', '-e', 'const str	Dead Code Obfuscation	FALSE	FALSE
Square root: 4	Dead Code Obfuscation	FALSE	FALSE
Error: Command '['node', '-e', 'const per	Dead Code Obfuscation	FALSE	TRUE
Current date and time: Tue Jun 11 2024	Dead Code Obfuscation	FALSE	TRUE
Maximum: 10	Dead Code Obfuscation	TRUE	FALSE
10 kilometers is 6.21371 miles	Dead Code Obfuscation	TRUE	TRUE
Hypotenuse: 5	Dead Code Obfuscation	TRUE	TRUE
Fixed number: 3.14	Dead Code Obfuscation	TRUE	TRUE
First character: J	Dead Code Obfuscation	TRUE	TRUE
Perimeter of rectangle: 30	Dead Code Obfuscation	TRUE	TRUE

#### **Anthropic Prompting**

- Anthropic is by the company of Claude 3
- Allows for advanced prompting
- Gives LLMs best chance of success





#### Your generated prompt

Click Start Editing to use this prompt, or discard it to generate a new one.

You will be given a simple JavaScript code snippet inside {{CODE}} tags:

<code> {{CODE}} </code>

You will also be given a specific obfuscation method to apply to the code inside {{OBFUSCATION\_METHOD}} tags:

<obfuscation\_method>
{{OBFUSCATION\_METHOD}}
</obfuscation\_method>

Your task is to obfuscate the provided JavaScript code using ONLY the specified obfuscation method. Do not make any other changes to the code except for applying the one obfuscation technique that was provided.

The valid obfuscation methods that may be specified are:

- Dead code insertion: Add code that has no effect on the program logic
- ∑ Check that the prompt makes sense when variables are replaced with representative values.

Discard



#### **Analysis**

#### Out of 373 JavaScript snippets:

- Correct implementation:
  - 248 passed and 125 failed.
- Correct output:
  - o 221 passed and 152 failed.
- Correct output & implementation:
  - 155 passed and 218 failed.
- Correct output but failed implementation:
  - o 66 passed
- Wrong output but correct implementation:
  - o 93 passed

#### **Findings**

- Output and implementation
  - More failed
- More accuracy with naming obfuscation
- Occasionally mixed obfuscation methods