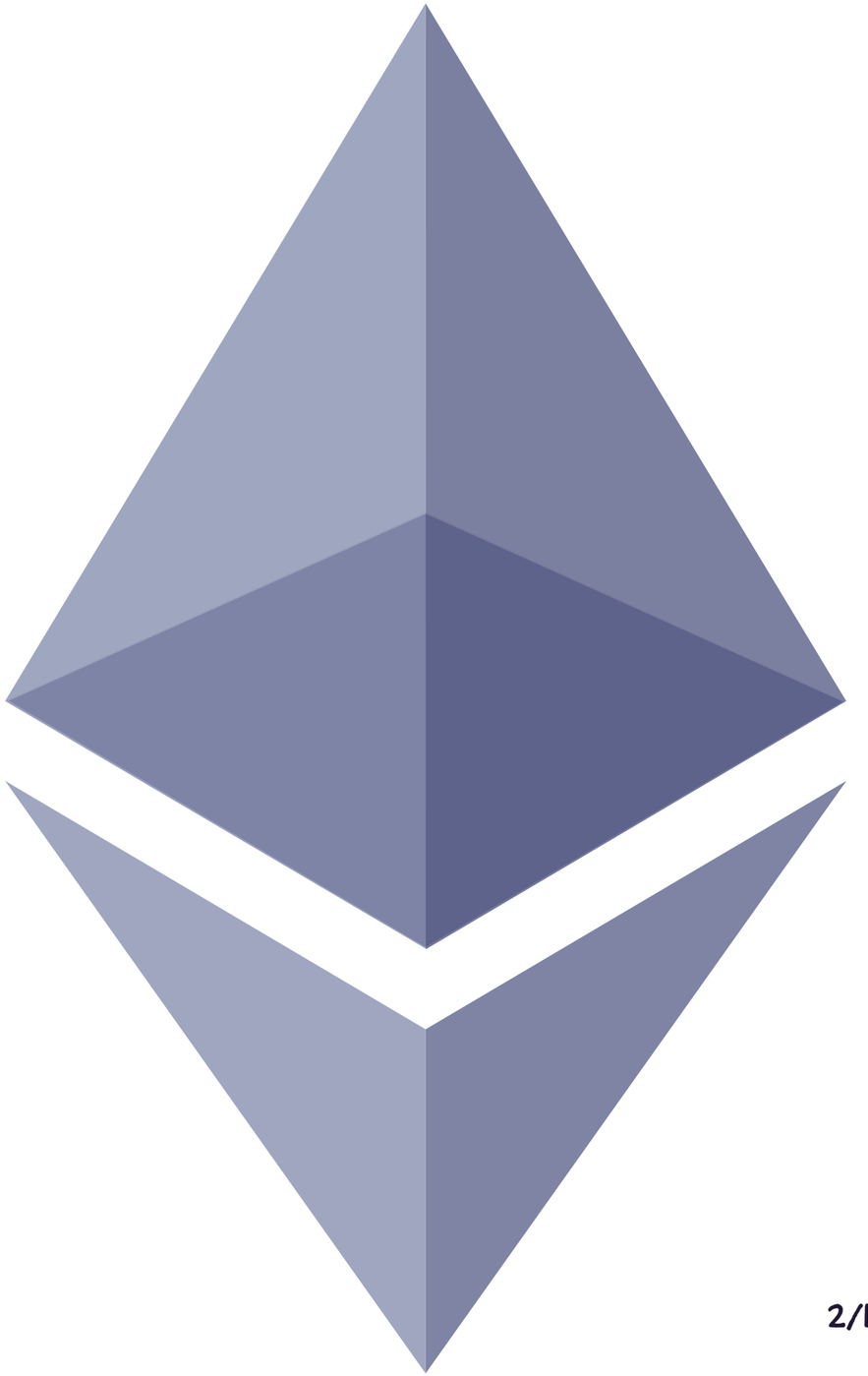
CS4215 Pontevedra

Source To EVM Compiler

- **Ethereum Virtual Machine** is part of the **Ethereum Network** that handles *smart contract* deployment and execution.
- Helps to maintain and update the states of the entire blockchain, containing millions of executable objects, each with its own permanent data store.
- Abstraction of computation and storage.



quasi-Turing complete

- Upper bound of execution processes by available **gas**.
- **gas** == 💰
- All program executions will halt.

EVM Architecture

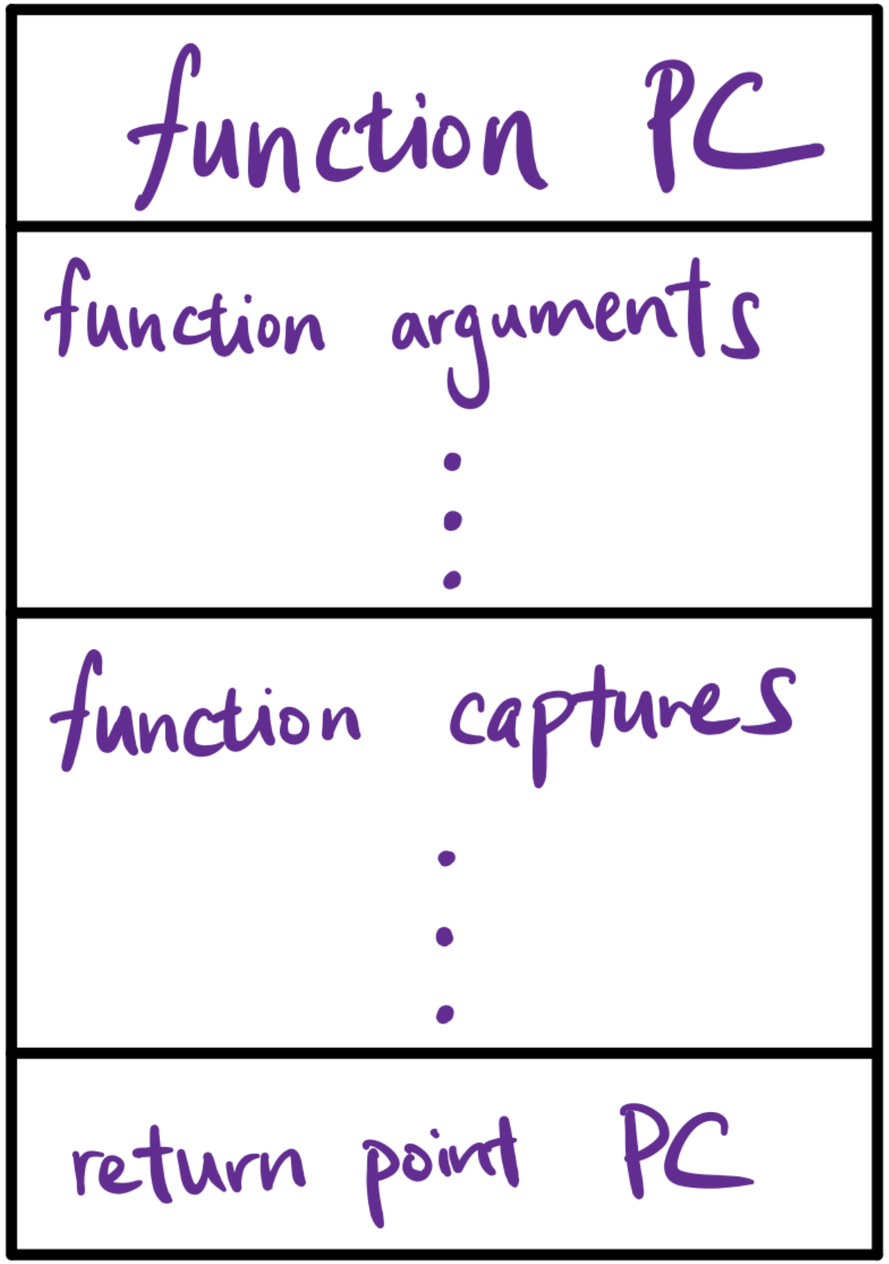
*applicable to this project

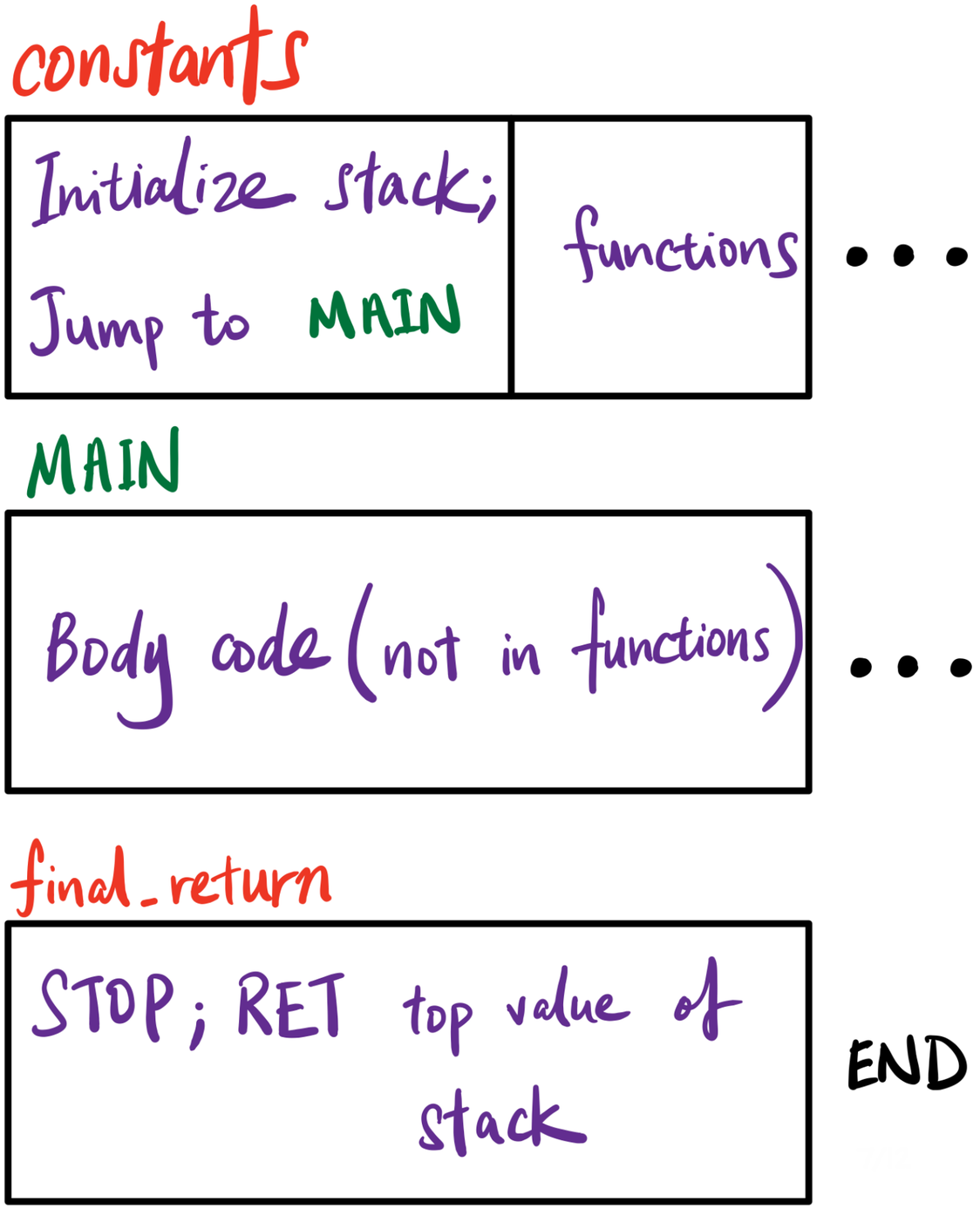
- Word Size: 256 bits
- Stores all in-memory values on a **STACK***.
- A volatile **MEMORY***.
- A permanent STORAGE.

Instruction Set (Bytecodes)

1. Arithmetic and bitwise logic operations
2. Stack, memory and storage access.
3. Control flow operations

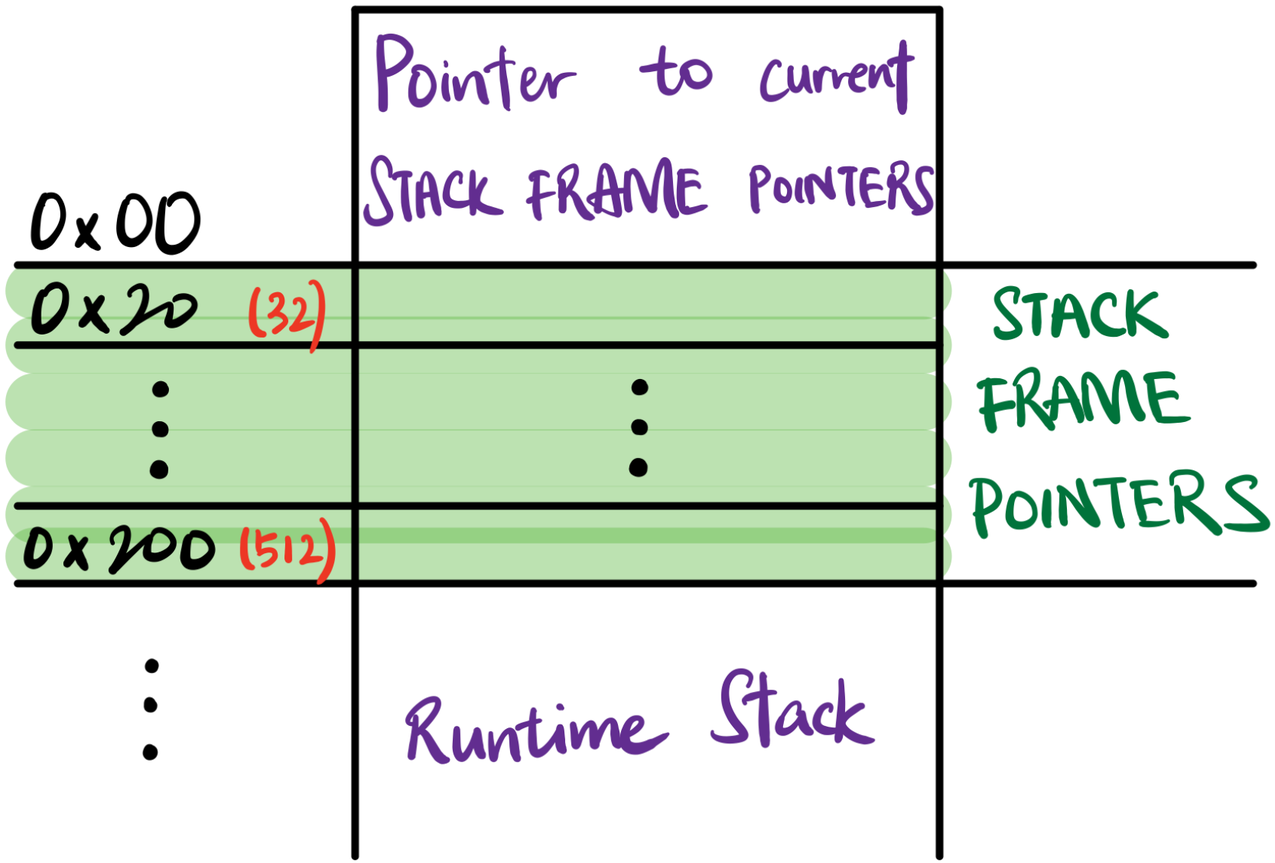
Bytecode	Stack Input	Stack Output	Effect
MLOAD	offset	value	reads a uint256 from memory
MSTORE	offset, value		writes a uint256 to memory
JUMPDEST			annotate possible jump destination
JUMP	destination		unconditional jump
JUMPI	destination, condition		jump if true
PC		PC	program counter
SWAPx	a ... b		swap the top of the stack with the x-th last element
DUP1	value	value, value	clones the last value on the stack
RETURN	offset, length		Halt execution and return output data


function PC
function arguments
function captures
return point PC


constants
Initialize stack;
Jump to MAIN
functions
MAIN
Body code (not in functions)
final_return
STOP; RET top value of stack
END

Example

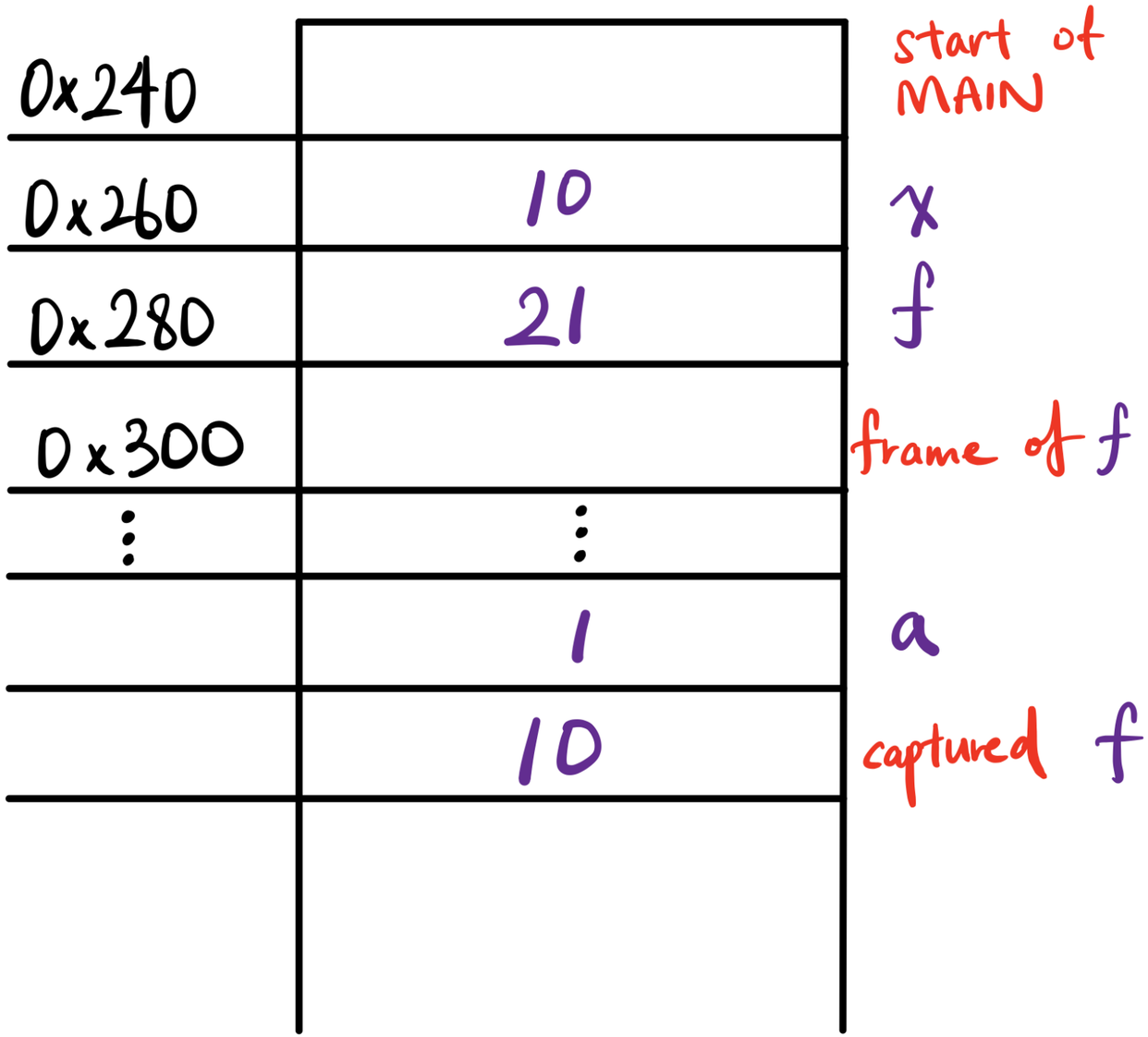
```
let x = 10;
function f(a) {
    return x + a;
}
f(1);
```



Example

```
let x = 10;
function f(a) {
    return x + a;
}
f(1);

// f(a) → f(a, x)
// must be a variable named `x`!
```



Demo

Highlights

- Functions
  - Named, anonymous
  - Nested
  - Recursion, mutual recursion, tail call optimisation
  - Functions as parameters and return values

Variable capture

- Create minimal closure by scanning free variables
- Convert free variables to arguments
- Capture variables by passing them as arguments
- All passed by value, no side effects
- Name of current function passed as additional variable
  - To support mutual recursion