



Cairo University

Faculty of computers and Artificial Intelligence

Cairo University



**CS213**

**Object Oriented Programming**

**Assignment 1**

**Task 4**

**Submitted to: Mohamed El-Ramly**

**Section: S2**

**Second level**

Name	ID
Abdulrahman Ashraf Mohamed Ahmed	20231094
Youssef Aboelyosre Afeed Ibrahim	20230475
Ahmed Osman Ali	20230801

# Vole Machine Simulator

## Detailed Class Design: Classes, Relationships, and Attributes:

### 1. Registers Class:

- **Attributes:**
  - registers: A map<string, string> storing 16 registers (labeled from "0" to "F"), each initialized to "0".
- **Methods:**
  - get (const string &reg): Returns the value of the specified register.
  - set (const string &reg, const string &value): Sets the value of the specified register.
  - display () const: Displays all registers and their values.
  - is\_valid\_register(char reg) const: Checks if the given register identifier is valid.

### 2. Memory Class:

- **Attributes:**
  - Memory: A **Memory** constructor initializes a memory map where each address, from **0x00** to **0xFF**, is set to the default value **"00"**. Each address is represented as a two-character hexadecimal string (e.g., **"00"**, **"01"**, ..., **"FF"**), and the **memory** map is populated with these addresses as keys, each associated with the value **"00"**. This setup ensures that all memory locations start with a known default state.
- **Methods:**
  - get (const string &addr): Retrieves the value at the specified memory address.
  - set (const string &addr, const string &value): Sets the value at the specified memory address.
  - display () const: Displays all memory locations and their values.
  - is\_valid\_memory(const string &addr) const: Checks if the given memory address is valid.

### 3. CU (Control Unit) Class:

- **Attributes:**

- registers: A reference to the Registers class.
- memory: A reference to the Memory class.
- **Methods:**
  - CU (Registers &regs, Memory &mem): Constructor that initializes the control unit with references to Registers and Memory.
  - execute\_instruction (const string &instruction): Executes a single instruction based on its opcode.
  - Specific opcode handling methods (e.g., rotatefunction, bitwise\_or, load\_memory\_to\_register, jump\_to, etc.).
  - halt () const: Halts the execution and displays the state of registers and memory.

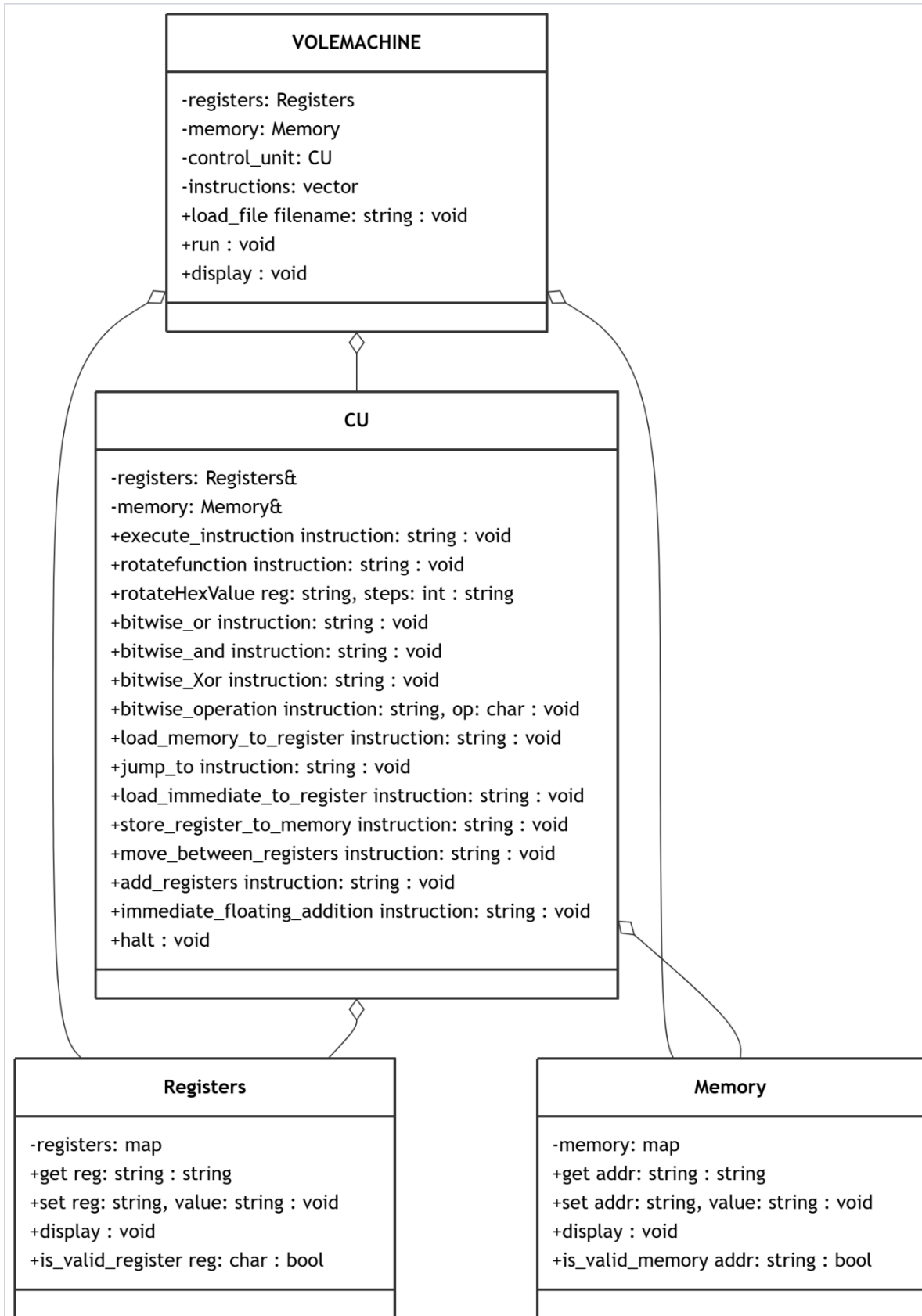
#### 4. VOLEMACHINE Class:

- **Attributes:**
  - registers: An instance of the Registers class.
  - memory: An instance of the Memory class.
  - control unit: An instance of the CU class, linked to Registers and Memory.
  - instructions: A vector<string> containing the list of instructions to be executed.
- **Methods:**
  - VOLEMACHINE (): Constructor that initializes Registers, Memory, and CU.
  - load\_file (const string &filename): Loads instructions from a file.
  - run (): Executes all instructions in sequence.
  - display () const: Displays the current state of registers and memory.

#### Relationships Between Classes:

- **VOLEMACHINE:**
  - Contains instances of Registers, Memory, and CU, linking them for operation.
- **CU (Control Unit):**
  - Operates on the Registers and Memory passed to it during instantiation, making it the processing core.
- **Registers and Memory:**
  - Independently maintain the state of the virtual machine's registers and memory, respectively, and provide methods for data access and modification.

# Vole Machine Design:



## work break-down table:

Name / ID	Work
<b>Abdulrahman Ashraf Mohamed - 20231094</b>	<ol style="list-style-type: none"><li>1. load_memory_to_register ()</li><li>2. rotate_function (), rotateHexValue ()</li><li>3. Load_immediate_to_register ()</li><li>4. Store_register_to_memory ()</li><li>5. bitwise_operation ()</li><li>6. display ()</li><li>7. menu</li><li>8. Report</li></ol>
<b>Youssef Aboelyosre Afeed - 20230475</b>	<ol style="list-style-type: none"><li>1. move_between_registers ()</li><li>2. add_registers ()</li><li>3. Immediate_floating_addition ()</li><li>4. bitwise or ()</li><li>5. OOP class design</li><li>6. Class Register, Class Memory</li><li>7. Getter and Setter</li><li>8. Github</li></ol>
<b>Ahmed Osman Ali - 20230801</b>	<ol style="list-style-type: none"><li>1. jump ()</li><li>2. bitwise xor ()</li><li>3. bitwise and ()</li><li>4. Jump_v2 ()</li><li>5. upload_file (), Validation file name</li><li>6. Validation instruction content</li><li>7. Halt ()</li></ol>

## Project on GitHub:

➔ Project link on GitHub:

<https://github.com/Youssefaboalyouser/VoleMachine> Assignment

## Screens of project:

**VoleMachine\_Assignment**
Public

Watch 1
Fork 0
Star 0

main
1 Branch
Tags

Add file
Code

**Youssefaboalyouser**
new version with reset functionality
dba5cef · 5 hours ago
6 Commits

.vscode	new version of implementation file	yesterday
README.md	Initial commit	4 days ago
VoleMachine.cpp	new version of implementation file	yesterday
VoleMachine.h	new version with reset functionality	5 hours ago
main.cpp	new version with reset functionality	5 hours ago
program2.txt	the implementation file for vole machine project	yesterday

README

# VoleMachine\_Assignment

*No description, website, or topics provided.*

Readme  
 Activity  
 0 stars  
 1 watching  
 0 forks  
[Report repository](#)

**Releases**  
 No releases published  
[Create a new release](#)

**Packages**  
 No packages published  
[Publish your first package](#)

**Contributors** 3
 

Youssefaboalyouser Youssef\_Aboalyous...  
 Abdulrahman-Ashraf161  
 Ahmd-OS

**Languages**  
 C++ 100.0%

Youssefaboalyouser / VoleMachine\_Assignment

Q

Type ↵ to search

+

⌵

🔄

🔍

📄

🌐

<>

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Files

main

Go to file

>

.vscode

README.md

VoleMachine.cpp

VoleMachine.h

main.cpp

program2.txt

VoleMachine\_Assignment / VoleMachine.h

Youssefaboalyouser

new version with reset functionality

dba5cef · 5 hours ago

History

Code

Blame

84 lines (72 loc) · 4.01 KB

Code 55% faster with GitHub Copilot

Raw

📄

📄

🔍

⌵

🔗

```
1  #ifndef VOLEMACHINE_H
2  #define VOLEMACHINE_H
3
4  #include <iostream>
5  #include <map>
6  #include <string>
7  using namespace std;
8
9  // Class to represent the CPU registers
10 class Registers
11 {
12 public:
13     Registers(); // Constructor to initialize registers
14     string get(const string &reg); // Retrieve the value of a specific register
15     void set(const string &reg, const string &value); // Set the value of a specific register
16     void display() const; // Display all registers and their values
17     bool is_valid_register(char reg) const; // Check if a given register is valid
18
19 private:
20     map<string, string> registers; // Map to store register names and values
21 };
22 --
```

Activity

All branches

All activity

All users

All time

Showing most recent first

new version with reset functionality

Youssefaboalyouser pushed 1 commit to `main` · 1f5a087...dba5cef · 5 hours ago

...

Add files via upload

Ahmd-OS pushed 1 commit to `main` · 23b7567...1f5a087 · yesterday

...

Add files via upload

Abdulrahman-Ashraf161 pushed 1 commit to `main` · 805e500...23b7567 · yesterday

...

new version of implemenation file

Youssefaboalyouser pushed 1 commit to `main` · 1cc1107...805e500 · yesterday

...

the implementation file for vole machine project

Youssefaboalyouser pushed 1 commit to `main` · d7d94e8...1cc1107 · yesterday

...

Initial commit

Youssefaboalyouser created `main` · d7d94e8 · 3 days ago

...

[Share feedback about this page](#)