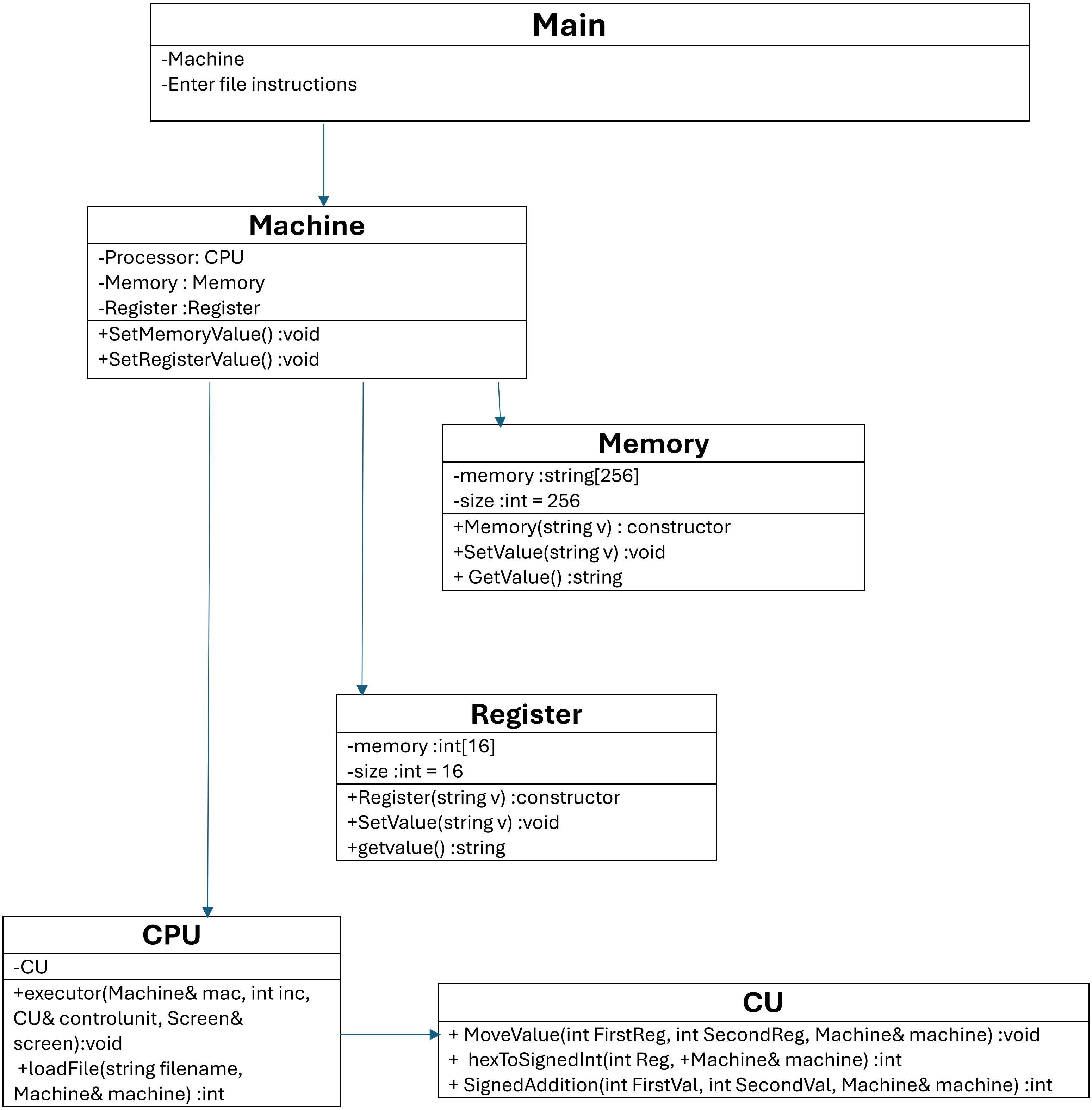
A1 Task 4

Group members:

Name	ID
➢Mazen Amr	➢20230307
➢Eyad Haitham	➢20230077
➢Yaseen Mohamed	➢20230468

Represented for Dr: Mohamed Elramly

Diagram:


Main
-Machine
-Enter file instructions
Machine
-Processor: CPU
-Memory : Memory
-Register :Register
+SetMemoryValue() :void
+SetRegisterValue() :void
Memory
-memory :string[256]
-size :int = 256
+Memory(string v) : constructor
+SetValue(string v) :void
+ GetValue() :string
Register
-memory :int[16]
-size :int = 16
+Register(string v) :constructor
+SetValue(string v) :void
+getvalue() :string
CPU
-CU
+executor(Machine& mac, int inc, CU& controlunit, Screen& screen):void
+loadFile(string filename, Machine& machine) :int
CU
+ MoveValue(int FirstReg, int SecondReg, Machine& machine) :void
+ hexToSignedInt(int Reg, +Machine& machine) :int
+ SignedAddition(int FirstVal, int SecondVal, Machine& machine) :int

```
+ decimalToHex(int sum) :string
 + hexToDecimal(int Reg, Machine& machine) :int
+ decimalToBinary(int decimal) :string
 + ExponentToBinary(int decimal) :string
+ BinaryToDecimal(string Binary) :int
 + BinaryFractionToDecimal(string Binary, Machine& machine):double
+ BinaryToFloating(string Binary, Machine& machine) :float
+ decimalFractionToBinary(double fractional) :string
+ FloatingToBinary(int newReg, float FloatNum, Machine& machine) :void
+ hex_to_dec(string hex) :int
+OP2(int target, string pattern, machine& mac) :void
```

Classes details:

Class name	Attributes	Functions
Register	value :string	SetValue(v :string): void
		getvalue(): string
Machine	Register :vector	getRegisters(): vector
	MemorySize :vector	getMemory(): vector
		SetMemoryvalue(index :int, value :string): void
		SetRegisterValue(index :int, value :string): void
CU		
		MoveValue(FirstReg :int, SecondReg :int , machine :Machine&): void MoveValue(int FirstReg, int SecondReg, Machine& machine) :void hexToSignedInt(int Reg, Machine& machine) :int SignedAddition(int FirstVal, int SecondVal, Machine& machine) :int decimalToHex(int sum) :string hexToDecimal(int Reg, Machine& machine) :int decimalToBinary(int decimal) :string ExponentToBinary(int decimal) :string BinaryToDecimal(string Binary) :int BinaryFractionToDecimal(string Binary, Machine& machine):double BinaryToFloating(string Binary, Machine& machine) :float decimalFractionToBinary(double fractional) :string

		FloatingToBinary(int newReg, float FloatNum, Machine& machine) :void hex_to_dec(string hex) :int OP2(int target, string pattern, machine& mac) :void
Screen	screenval :vector	addToScreen(value :string): void
		clearScreen(): void
		printScreen(): void
Memory	size :string	SetValue(v :string) :void
	value :string	Getvalue() :string
CPU	CU :as a member	executor(Machine& mac, int inc, CU& controlunit, Screen& screen):void loadFile(string filename, Machine& machine) :int

Classes overview:

1. Register

- **Purpose:** Represents a register, which is a small, fast storage location in a CPU used to hold data.

- **Attributes:**
  - value (string): Holds the data of the register.
- **Methods:**
  - Register(string v): Constructor to initialize the register with a specific value.
  - SetValue(string v): Sets the value of the register.
  - getvalue(): Returns the current value of the register.

2. Memory

- **Purpose:** Represents a memory cell in a machine's memory.
- **Attributes:**
  - value (string): Holds the value stored in the memory cell.
- **Methods:**
  - Memory(string v): Constructor to initialize the memory cell with a specific value.
  - SetValue(string v): Sets the value of the memory cell.
  - GetValue(): Returns the value stored in the memory cell.

3. Machine

- **Purpose:** Represents a machine (or simulated computer), which includes a collection of registers and memory cells.
- **Attributes:**
  - Registers (vector<Register>): A list of registers.
  - MemorySize (vector<Memory>): A list of memory cells.
- **Methods:**
  - Machine(): Constructor to initialize the machine, setting up registers and memory.
  - getRegisters(): Returns a reference to the vector of registers.

- getMemory(): Returns a reference to the vector of memory cells.
- SetMemoryValue(int index, string value): Sets the value of a memory cell at a specific index.
- SetRegisterValue(int index, string value): Sets the value of a register at a specific index.

4. CU (Control Unit)

- **Purpose:** Represents the control unit of a CPU, responsible for executing instructions and managing data flow within the machine.
- **Methods:**
  - MoveValue(int FirstReg, int SecondReg, Machine& machine): Moves data from one register to another.
  - hexToSignedInt(int Reg, Machine& machine): Converts a hexadecimal value in a register to a signed integer.
  - SignedAddition(int FirstVal, int SecondVal, Machine& machine): Adds two signed integers and returns the result.
  - decimalToHex(int sum): Converts a decimal integer to a hexadecimal string.
  - hexToDecimal(int Reg, Machine& machine): Converts a hexadecimal value in a register to a decimal integer.
  - decimalToBinary(int decimal): Converts a decimal integer to a binary string.
  - ExponentToBinary(int decimal): Converts the exponent part of a decimal number to a binary string (likely for floating-point operations).
  - BinaryToDecimal(string Binary): Converts a binary string to a decimal integer.
  - BinaryFractionToDecimal(string Binary, Machine& machine): Converts the fractional part of a binary number to a decimal (likely for floating-point values).

    - float BinaryToFloating(string Binary, Machine& machine): Converts a binary string to a floating-point number.
    - decimalFractionToBinary(double fractional): Converts a fractional decimal number to binary.
    - FloatingToBinary(int newReg, float FloatNum, Machine& machine): Converts a floating-point number to binary and stores it in a register.
    - OP2(int target, string pattern, Machine& mac): Sets a register's value to a specific pattern (likely an opcode or binary value).
    - hex_to_dec(string hex): Converts a hexadecimal string to a decimal integer.

5. Screen

- **Purpose:** Manages the output display for the simulated computer, showing the current contents of registers and memory.
- **Attributes:**
    - screenval (vector<string>): Stores the screen values (e.g., output, instructions).
- **Methods:**
    - addToScreen(string value): Adds a value to the screen.
    - clearScreen(): Clears the screen.
    - printScreen(): Prints the screen content (likely the values on the screen).
    - displayMemory(Machine& mac): Displays the current memory content of the machine.
    - displayRegister(Machine& mac): Displays the current values of the registers.
    - displayScreen(): Displays the screen's content.

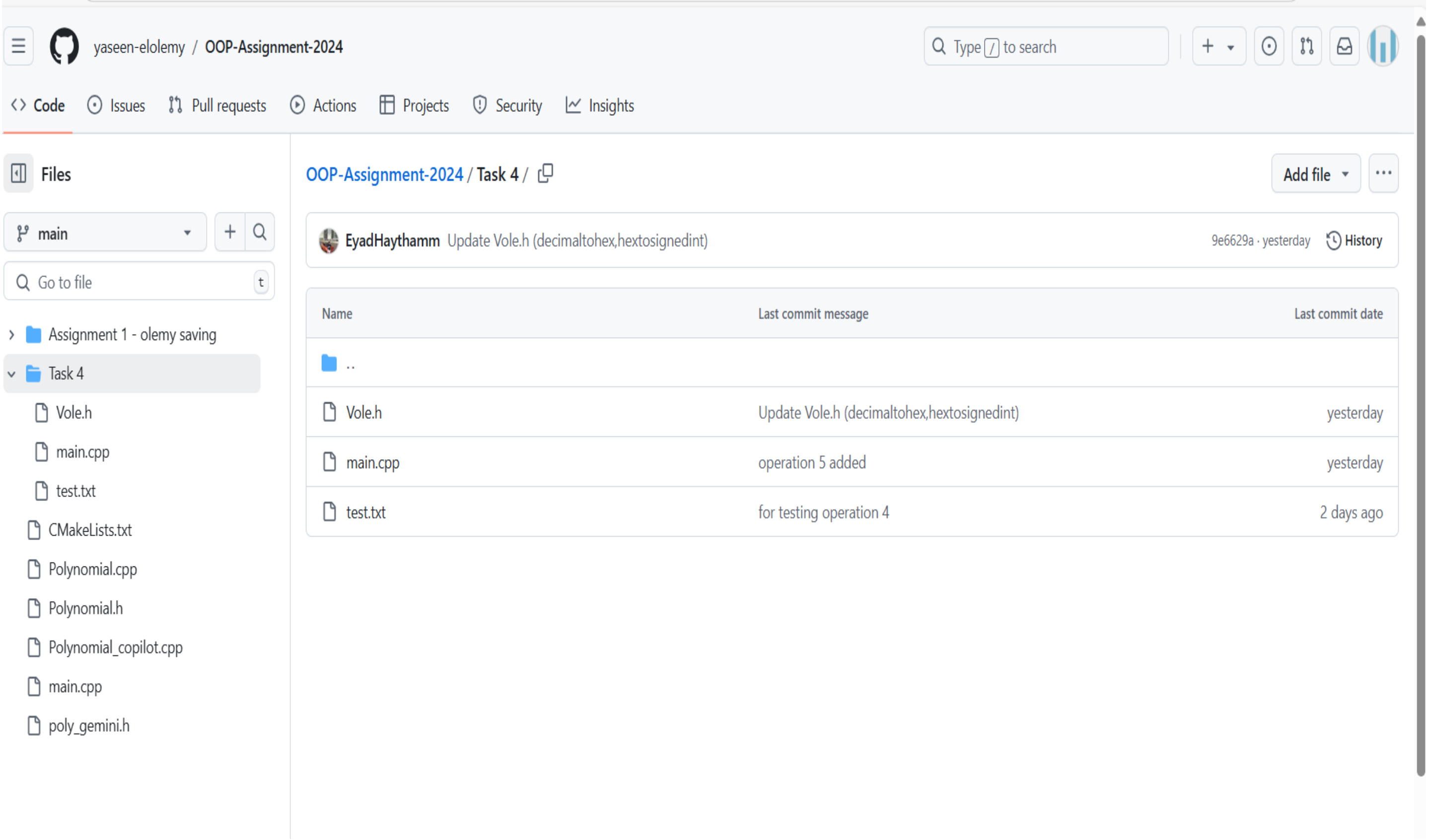
6. CPU

- **Purpose:** Represents the CPU, which uses the control unit (CU) to execute instructions and interact with the machine.
- **Attributes:**
  - controlunit (CU): An instance of the control unit that processes instructions.
- **Methods:**
  - CPU(): Constructor to initialize the control unit.
  - executor(Machine& mac, int inc, CU& controlunit, Screen& screen): Executes a program or instruction on the machine.
  - loadFile(string filename, Machine& machine): Loads a program from a file into the machine

<u>Work break-down table</u>:

Task done	Responsible person	Person ID	Status
1 RXY	Yaseen mohamed	20230468	Completed
2 RXY	Yaseen mohamed	20230468	Completed
3 RXY	Yaseen mohamed	20230468	Completed
3 R00	Mazen amr	20230307	Completed
4 0RS	Eyad haitham	20230077	Completed
5 RST	Eyad haitham	20230077	Completed
6 RST	Eyad haitham	20230077	Completed
B RXY	Mazen amr	20230307	Completed
C 000	Mazen amr	20230307	Completed
Menu	Yaseen mohamed	20230468	Completed
Loop implementation	Yaseen mohamed	20230468	Completed
Register,Memory,CPU&CU(classes)	Eyad haitham	20230077	Completed
Machine (class)	Mazen amr	20230307	Completed
Machine (class)	Eyad haitham	20230077	Completed
Screen (class)	Yaseen mohamed	20230468	Completed
Report	Mazen amr	20230307	Completed

<u>GitHub project:</u>



OOP-Assignment-2024 / Task 4 /

EyadHaythamm Update Vole.h (decimaltohex,hextosignedint) 9e6629a · yesterday

Name	Last commit message	Last commit date
..		
Vole.h	Update Vole.h (decimaltohex,hextosignedint)	yesterday
main.cpp	operation 5 added	yesterday
test.txt	for testing operation 4	2 days ago

Input form:

```
0x2 0x0 0x05

0x2 0x1 0x00

0x2 0x2 0x01

0x5 0x1 0x12

0xB 0x1 0x0C

0xB 0x0 0x06

0x3 0x1 0x00

0xC 0x0 0x00
```

Expected output:

5

Note: Hexa input must be capitalized