

# CSE241 – OOP (Fall 2025)

## Homework #1

**Due Date: 15.10.2025 23:59**

**Hand-in Policy:** Source code and any documentation should be submitted online as a single .zip or .rar file with naming convention STUDENTID\_LASTNAME\_FIRSTNAME\_H4.ZIP via Teams by the submission deadline. No late submissions will be accepted.

**Collaboration Policy:** No collaboration is permitted. Any cheating (copying someone else's work in any form) will result in a grade of -100 for the first offense and -200 for the subsequent attempts.

**Grading:** Each homework will be graded on the scale of 100. Unless otherwise noted, the questions/parts will be weighed equal.

---

You will write a C++ program that will simulate a very simple CPU. Your CPU has only 6 registers (R1, R2, R3, R4, R5, R6) and no other memory. Your CPU has a small set of instructions as described below.

- Move instructions

```
MOV reg, reg  
MOV reg, constant
```

For example, `MOV R1, R2` copies the value of register 1 to register 2 and `MOV R1, 100` puts the value of 100 to register 1

- Arithmetic instructions

```
ADD reg, reg  
ADD reg, constant  
SUB reg, reg  
SUB reg, constant
```

For example, `ADD R1, R2` adds the value of register 2 to register 1. `ADD R1, 100` adds the value of 100 to register 1. `SUB R1, R2` subtracts the value of register 2 from register 1 and puts the result into register 1.

- Jump instructions

```
JMP reg, lineAddress  
JMP lineAddress
```

For example, `JMP R1, 32` jumps to line 32 of the program if the value of R1 is zero. `JMP 23` jumps to line 23 directly.

- Input/output instructions

```
PRN reg
PRN constant
INP reg
```

For example, `PRN R3` will print the value of register 3 to the screen, after each print a new line should be inserted. `INO R1` will get a character from the keyboard and store its ASCII value in register 1.

- Other instructions

```
HLT
```

Halts the program and prints on the screen the contents of all registers.

Following is a program file that prints numbers from 10 down to 0 to the screen

```
MOV R1, 10      ; line 1, load value 10 to the register 1
PRN R1         ; line 2, print register 1
SUB R1, 1       ; line 3, decrement R1
JMP R1, 6       ; line 4, if R1 is 0 then go to line 6
JMP 2           ; line 5, go to line 2
HLT             ; finish the program
```

As seen above, you may have comments after each instruction after the ';' sign. Each line of your program file should contain a single instruction, so the line numbers will correspond to the instruction numbers.

Your program will run using command line parameters. The format for the command line parameters is as follows:

```
yourProg filename option
```

`yourProg` is the name of your executable file (as compiled by your C++ compiler), `filename` is the text file that contains your simple CPU instructions, `option` a number and the defines the how your program runs as follows

- `option = 0`: program will run and finish by executing each instruction.
- `option = 1`: program will execute each instruction after displaying the instruction first. It also will print the contents of all the registers such as

```
MOV R1, 10 - R1=10, R2=0, R3=0, R4=0, R5=0, R6=1
```

Important Considerations:

- Write a CPU program that adds the squares of integers between 1 and 10. The result should be printed on the screen. Include your CPU program with your submission.
- Your program should handle error cases such as syntax errors in the input files. You should print an error message on the screen and halt the program if you detect an error in the input.
- With your submission, include the results of a few runs of your program with different programs and run options.
- Do not use any functions from the standard C library (like `printf`), you will use << and >> operators to print and write strings.
- You will use C++ string class to manipulate your strings.
- Your program should have only functions and no classes.
- Do not forget to “indent” your code and provide comments.