Gebze Techical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2016
Homework # 3
Our CPU in C++
Due date Oct 27th 2016

This homework will be very similar to HW1, only it will use OOP techniques.

First, you will design and implement a class for a CPU that has 5 registers (R1, R2, R3, R4, R5) and no other memory. Your CPU also has a special register PC (program counter) that keeps the current program line number that it executes. All registers are initialized to zero (PC is 1) when the CPU objects are formed. Your CPU class should have functions for

- Appropriate constructor(s)
- Setters and getters for all registers and PC
- **`print`** function that prints all the registers and the PC
- **`halted`** function returns true if the last instruction executed halted the CPU.
- **`execute`** function takes a string that contains an instruction line, executes it and advances the PC accordingly. For example, **`myCPU.execute("MOV R1, 100")`** puts the value of 100 to the register 1.

The instruction set for your CPU class will be the exactly the same as HW1.

Second, you will design and implement another class for a CPU program. Your **`CPUProgram`** will have the following member functions:

- **`ReadFile`** function takes a string file name as a parameter, reads the program file and stores all the program instructions.
- A constructor that takes a file name as parameter and calls the functions **`readFile`**.
- **`getLine`** function takes an integer as the parameter and returns a string that contains the program line that corresponds to the integer.
- **`size`** function returns the number of lines read from the file.

Your main function will make one **`CPUProgram`** object and one **`CPU`** object. The file name and the run time options will come from the command line as in HW1. You will use getLine function of **`CPUProgram`** to get the program lines and use the execute function of CPU objects to execute the lines. In other words, there will be a loop in your main function such as below

```
// some code here
CPUProgram myProg( /* params here */);
CPU myCPU( /* params here */);
// some code here
while(!CPU.halted()){
      // some code here
      string instruction = myProg.getLine(myCPU.getPC());
      myCPU.execute(instruction);
      // some code here
}
```

Important Notes:

- Your command line parameters will be the same as HW1
- Use the square adding program from HW1 to test your new classes.
- 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.
- Use all the OOP principles that we learned in the class.
- **Use separation of interface and implementation. In other words, you will have at least 3 .cpp files (driver, CPU, and CPUProgram) and 2 header files.**
- Do not forget to indent your code and provide comments.
- You should submit your work to the moodle page. You should strictly follow the submission instructions.