Gebze Techical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2016
Homework # 4
Memory and Computer in C++
Due date Nov 6th 2016

This homework will be very similar to HW2, only it will use OOP techniques such as composition. You will design and implement new classes.

First, you will design and implement a class for a **`Memory`**. Your memory can hold 50 unsigned integers as in HW2. The addreses start from 0. Your memory class has getter and setter functions for memory locations.

Modify your CPU class of HW3 so that your CPU can now handle memory instructions as listed in HW2. Modify your execute function of your CPU class so that it takes an intruction as a string and a memory object. For example,

```
myCPU.execute("MOV R1, #45", myMemory)
```

moves the value of R1 into the memory location 45 in **`myMemory`**.

Write another class named **`Computer`**. This class will have objects of **`CPU`**, **`Memory`** and **`CPUProgram`** as its data members. It has all necessary constructors and setter/getters. The class **`Computer`** has a function named execute. It then executes loads program and executes it on the **`CPU`** using computers memory.

Important Notes:

- Your command line parameters will be the same as HW2
- You should test your classes with the main function attached with this HW and attach the results. You will also write and test other main functions.
- Use the programs from HW1 and HW2 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.
- 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.

```
#include "requiredIncs.h" int main(int argc, char** argv){

////////////////////////////////////////////////////////////////////////// //command
line parameters const char* filename = argv[1]; int option = atoi(argv[2]);
//////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////
//Testing class Memory
Memory myMemory(option);

//index, value myMemory.setMem(0, 100); cout
<< myMemory.getMem(0) << endl; //should print
in a way that similar to this:
//Memory Values:
//[0] -> 100
//[1] -> 0
//[2] -> 0
//.
//.
//[49] -> 0 myMemory.printAll();
//////////////////////////////////////////////////////////////////////////


//////////////////////////////////////////////////////////////////////////
//Testing class CPU  CPU
myCPU(option);
myCPU.execute("MOV #0, R1", myMemory); myCPU.execute("MOV
R1, #1", myMemory);

//should print in a way that similar to this:
//CPU Register Values:
//[0] -> 100
//[1] -> 0
//[2] -> 0
//[3] -> 0 //[4]
-> 0
myCPU.print();

//should print in a way that similar to this:
//Memory Values: //[0]
-> 100
//[1] -> 100
//[2] -> 0
//.
//.
//[49] -> 0 myMemory.printAll();
//////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////
//Testing class CPUProgram  CPUProgram
myCPUProgram(option);
myCPUProgram.ReadFile(filename); cout <<
myCPUProgram.getLine(0) << endl; cout <<
```

```
    myCPUProgram.getLine(myCPUProgram.size(
    ) - 1) << endl;
    ////////////////////////////////////////////////////////////////////////

    ////////////////////////////////////////////////////////////////////////
    //Testing class Computer
    Computer myComputer1(myCPU, myCPUProgram, myMemory, option);
    Computer myComputer2(option); myComputer2.setCPU(
    myComputer1.getCPU() );
    myComputer2.setCPUProgram(myComputer1.getCPUProgram() );
    myComputer2.setMemory(myComputer1.getMemory() ); myComputer2.execute();
    ////////////////////////////////////////////////////////////////////////
    return 0;
}
```