Gebze Techical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2016
Homework # 6
Dynamic Memory and Classes
Due date
Nov 30th 2016

You will rewrite your CPUProgram of HW5 class so that uses dynamic memory. Your new CPUProgramDyn class has the same public member functions and overloaded operators. Therefore the customers of your class will not see any difference. However, your new class will not use any vectors and it has to allocate and deallocate its memory as required. Additionally you should follow the following rules

- Use your own namespace for this class.
- There is no maximum number of lines limit.
- All error checking should be done, for example [] should exit the program if out of index error is encountered.
- Do not forget to indent your code and provide comments.
- You should submit your work to the moodle page. Use the following main function as one of the test cases. You will also provide other test cases of your own.

```
#include "requiredIncs.h"

void testFunc(CPUProgramDyn myCPUProgram)
{
    myCPUProgram.ReadFile(filename);

    cout << myCPUProgram[0] << endl;
    cout << myCPUProgram[myCPUProgram.size() - 1] << endl;

    //op +
    cout << ((myCPUProgram + "MOV R1, #45")[myCPUProgram.size() - 1]) << endl;

    //op +=
    myCPUProgram += "MOV R2, #50";
    cout << myCPUProgram[myCPUProgram.size() - 1] << endl;

    //op + <<
    CPUProgramDyn myOtherCPUProgram(option);
    myOtherCPUProgram.ReadFile(filename);

    cout << (myCPUProgram + myOtherCPUProgram) << endl;

    //op COMP --
    cout << (myCPUProgram == myOtherCPUProgram ? "DONE" : "FAIL") << endl;
    cout << (myCPUProgram <= myOtherCPUProgram ? "DONE" : "FAIL") << endl;
    cout << (myCPUProgram > myOtherCPUProgram ? "FAIL" : "DONE") << endl;

    --myOtherCPUProgram;

    cout << (myCPUProgram != myOtherCPUProgram ? "DONE" : "FAIL") << endl;
    cout << (myCPUProgram >= myOtherCPUProgram ? "DONE" : "FAIL") << endl;
    cout << (myCPUProgram < myOtherCPUProgram ? "FAIL" : "DONE") << endl;

    //op ()
    cout << myCPUProgram(5, 10) << endl;

    //error check
    cout << myCPUProgram[myCPUProgram.size()] << endl;

    myCPUProgram += "";
    cout << myCPUProgram[myCPUProgram.size() - 1] << endl;
    //////////////////////////////////////////////////////////////////////////

}



int main(int argc, char** argv){

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

    //////////////////////////////////////////////////////////////////////////
    //Testing class CPUProgramDyn
    CPUProgramDyn myProg(option);

    testFunc(myProg)

    // Test again
    testFunc(myProg)

    // Test with a copy
    CPUProgramDyn myProgCopy(myProg);
    testFunc(myProgCopy);

    return 0;
}
```