ITU Computer Engineering Department
BLG252E Object Oriented Programming
3rd Homework

Asst. Prof. Dr. Gülşen Cebiroğlu Eryiğit
Asst. Prof. Dr. Sanem Sarıel Talay
Res. Asst. Doğan Altan
Res. Asst. Aycan Atak

Due Date: May 9, 2012 23.59 PM

In this homework, you will implement a generic Matrix class. On this Matrix class, some arithmetic operations can be applied. Some rules must be taken into consideration and if the given rules are not met, error messages must be thrown.

- To sum two matrix objects, their sizes (m x n) must be equal.
- To divide two matrix objects, their sizes (m x n) must be equal and the divisor matrix should not include any elements which is 0.
  You can simply apply divide operator as (Assume two 2 x 3 matrices);
  $a_{11}\ a_{12}\ a_{13}$ $b_{11}\ b_{12}\ b_{13}$ $a_{11}/b_{11}\ a_{12}/b_{12}\ a_{13}/b_{13}$
  $a_{21}\ a_{22}\ a_{23}$ $b_{21}\ b_{22}\ b_{23}$ $a_{21}/b_{21}\ a_{22}/b_{22}\ a_{23}/b_{23}$
- A matrix object cannot be compared or assigned to another object if their sizes are different.
- "Index out of bounds" message should be printed if one tries to access matrix with the non-existing index values.

A test program is given below.

```
int main()
{
	srand(time(NULL));
	Matrix<int> m1(3,5);  // creating some objects
	Matrix<int> m2(3,5);  // matrices' elements are assigned randomly from 0 to 10
	Matrix<double> m3(5,5);
	Matrix<double> m4(5,6);

	try{
		cout << m1.getElement(3,6) << endl;   // trying to get the element at(3,6)
	}
	catch(const string & err_msg)
	{
		cout << err_msg << endl;
	}

	cout << "Printing m4" << endl;
	m4.print();                              // printing m4

	try{
		Matrix<double> m6 = m4 / m3;        // trying to divide two matrices
	}
```

```
	catch(const string & err_msg)
	{
		cout << err_msg << endl;
	}

	cout << "Printing m1" << endl;
	m1.print();

	if(m1.contains(4))	// checking if the matrix has an element with value 4
		cout << "Matrix contains the element" << endl;
	else
		cout << "Matrix does not contain the element" << endl;

	Matrix<int> m5 = m2;
	cout << "Printing m5" << endl;
	m5.print();

	try{
		--m1;		// decrement m1's matrix elements by 1
		m5 = m1 + m2;	// sum m1 and m2 object's matrices and assign result to m5
		++m3;		// increment m1's matrix elements by 1
		m2 = m5 / m1;	// divide m5 and m1 object's matrices and assign result to m2
		if(m5 == m1)	// comparing two objects
			cout << "Objects are equal" << endl;
		else
			cout << "Objects are not equal" << endl;
	}
	catch(const string & err_msg)
	{
		cout << err_msg << endl;
	}

	cout << "Printing m5" << endl;
	m5.print();
}
```

For any questions, please send Doğan Altan (daltan@itu.edu.tr) an e-mail.

Submission Procedure:

1. Your source code should be named as “studentID_hw3.cpp”.
2. Make sure you write your name and number to all the header files of your project with the following format.

```
/*
 *
 *  BLG252E
 *  2012 Spring
 *  3rd Homework
 *
 */

/*
 * @Author
 * Student Name: !! enter here !!
 * Student ID  : !! enter here !!
 * Date:
 */
```

3. Make sure that GNU C++ Compiler (g++) compiles your project and the application runs in Unix smoothly. This is important because we will evaluate your homework in Unix using g++.
4. Use comments wherever necessary in your code to explain what you did.
5. After you make sure that everything is compiled smoothly, archive all files into a zip file. Submit this file through www.ninova.itu.edu.tr. Ninova enables you to change your submission before the submission deadline.

Academic dishonesty including but not limited to cheating, plagiarism, collaboration is unacceptable and subject to disciplinary actions. Any student found guilty will get grade F.