MASTER ELECTRONIC DESIGN

Matrix Homework

Homework2

Miguel Tlapa Juárez

2/11/2014

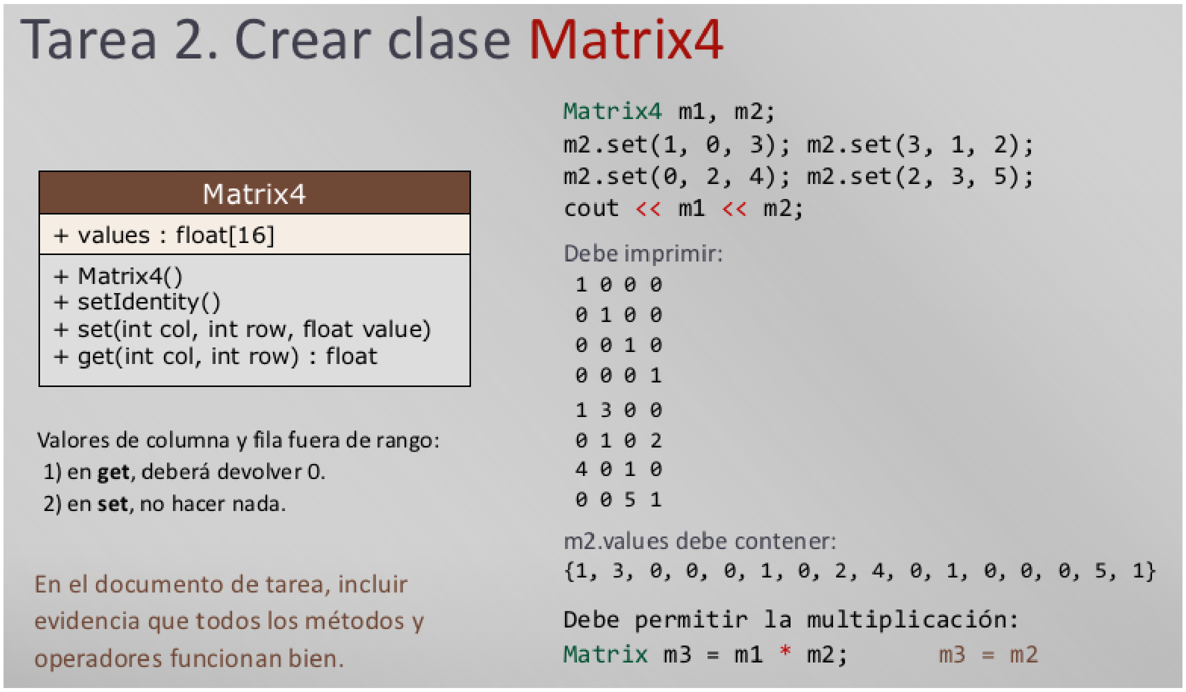
This document describes the system architecture and design about the body controller module, it’s have block diagram and flowchart to describe software and hardware architecture.

Revision History

Date	Revision Number	Author/Editor	Modifications
January 2014	0.1	Miguel Tlapa	Created file

Disclaimers

1. Explanation



Tarea 2. Crear clase Matrix4

Matrix4
+ values : float[16]

+ Matrix4()
+ setIdentity()
+ set(int col, int row, float value)
+ get(int col, int row) : float

Valores de columna y fila fuera de rango:
1) en **get**, deberá devolver 0.
2) en **set**, no hacer nada.

En el documento de tarea, incluir evidencia que todos los métodos y operadores funcionan bien.

```
Matrix4 m1, m2;
m2.set(1, 0, 3); m2.set(3, 1, 2);
m2.set(0, 2, 4); m2.set(2, 3, 5);
cout << m1 << m2;
```

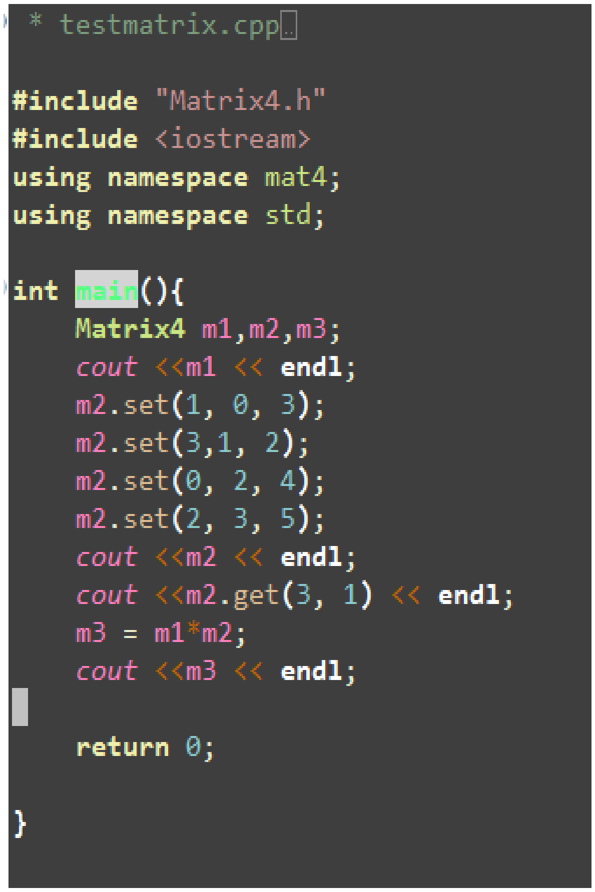
Debe imprimir:

```
 1 0 0 0
 0 1 0 0
 0 0 1 0
 0 0 0 1
 1 3 0 0
 0 1 0 2
 4 0 1 0
 0 0 5 1
```

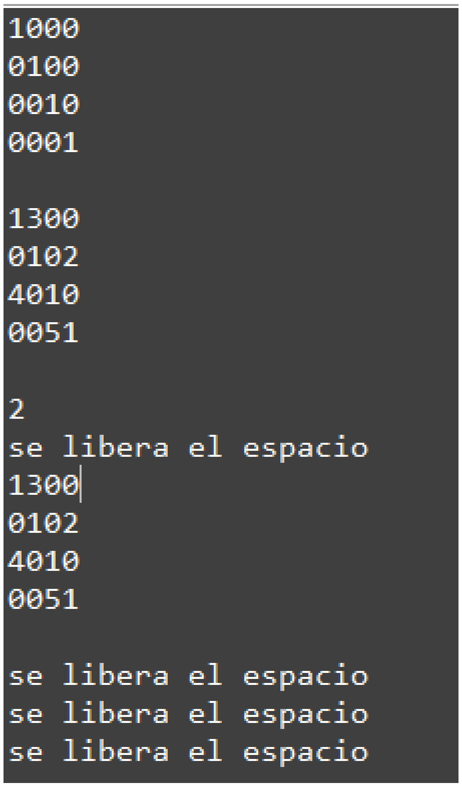
m2.values debe contener:

```
{1, 3, 0, 0, 0, 1, 0, 2, 4, 0, 1, 0, 0, 0, 5, 1}
```

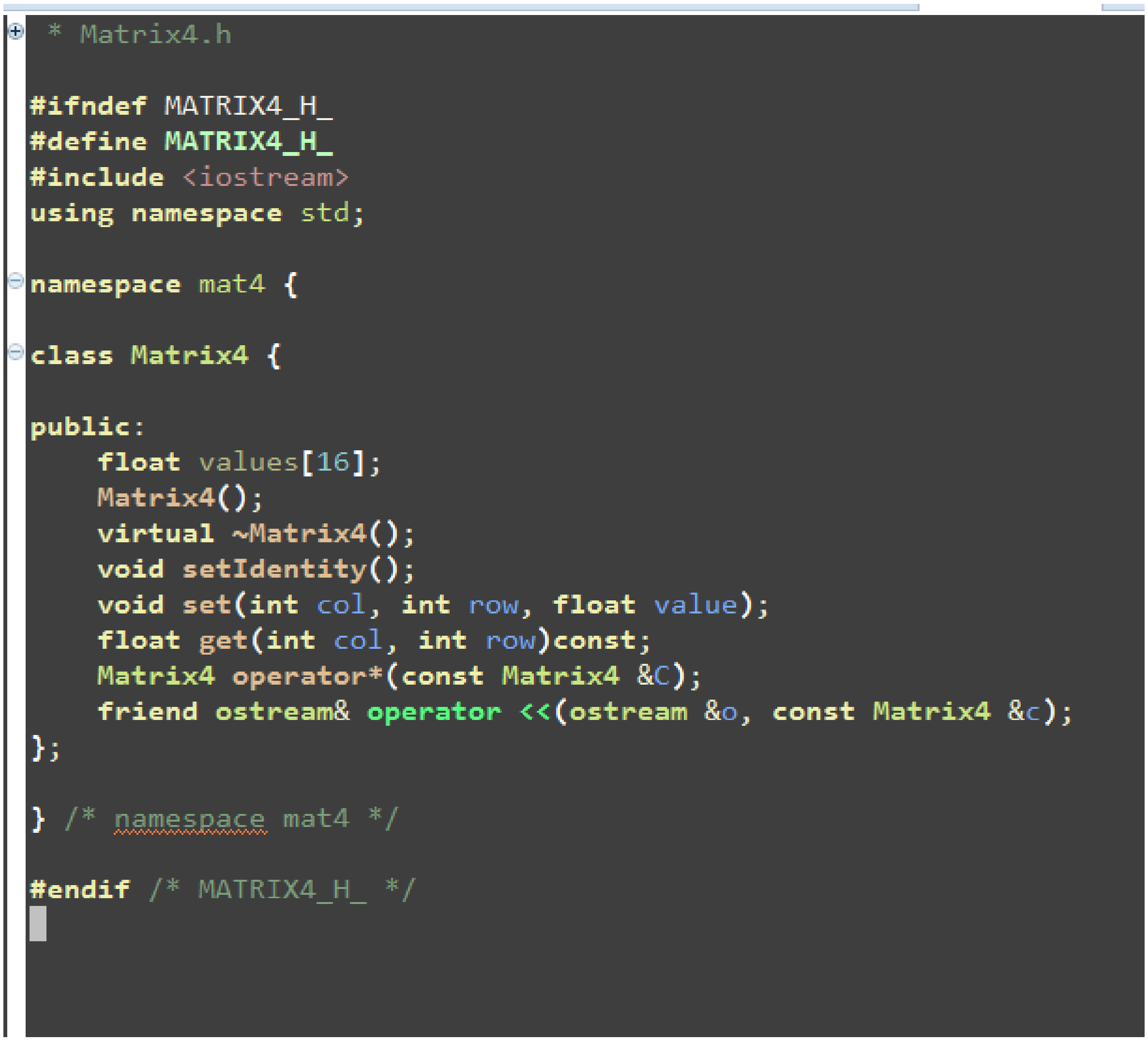
```
Debe permitir la multiplicación:
Matrix m3 = m1 * m2;        m3 = m2
```



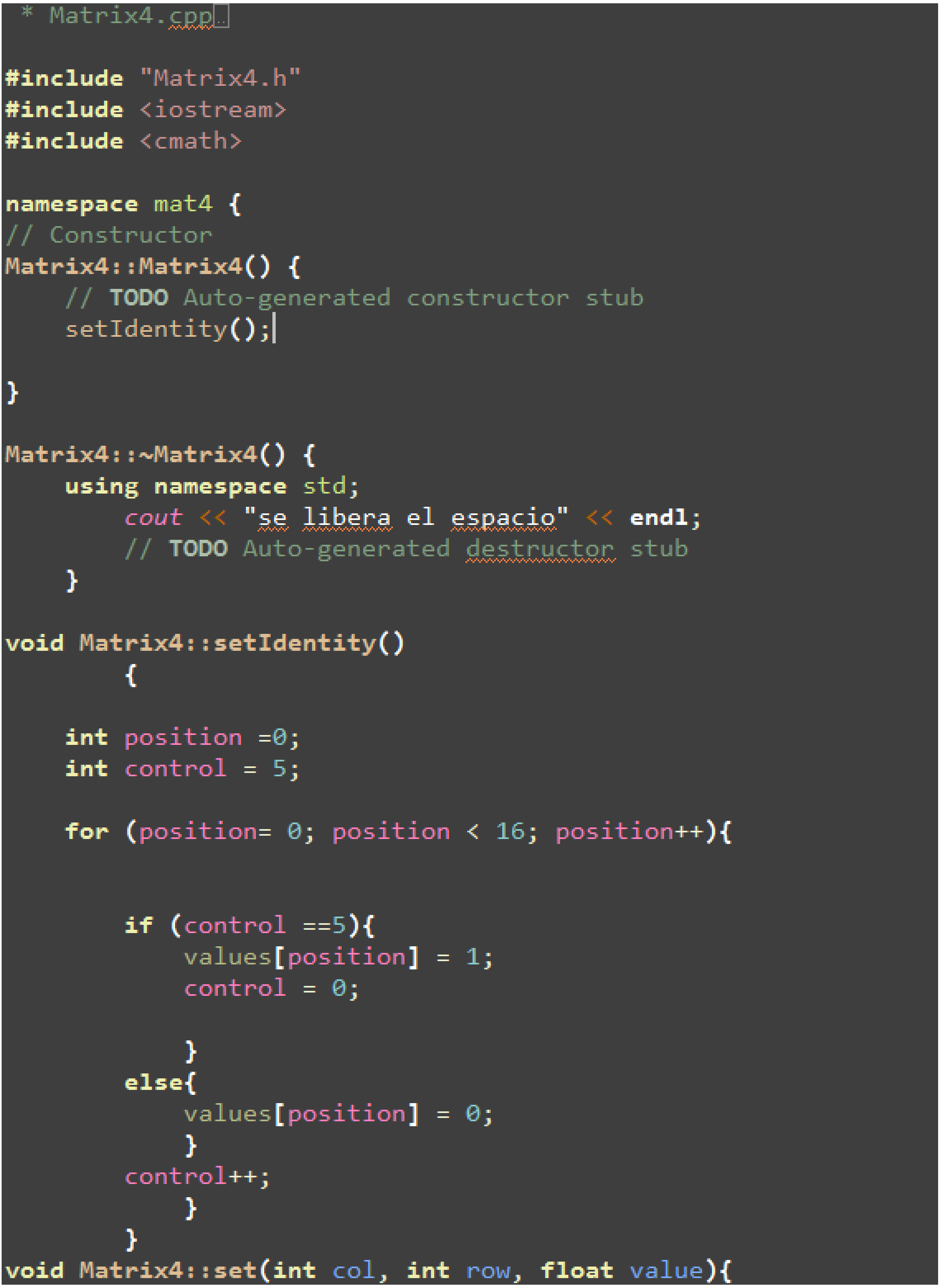
```
* testmatrix.cpp

#include "Matrix4.h"
#include <iostream>
using namespace mat4;
using namespace std;

int main(){
    Matrix4 m1,m2,m3;
    cout <<m1 << endl;
    m2.set(1, 0, 3);
    m2.set(3,1, 2);
    m2.set(0, 2, 4);
    m2.set(2, 3, 5);
    cout <<m2 << endl;
    cout <<m2.get(3, 1) << endl;
    m3 = m1*m2;
    cout <<m3 << endl;

    return 0;

}
```



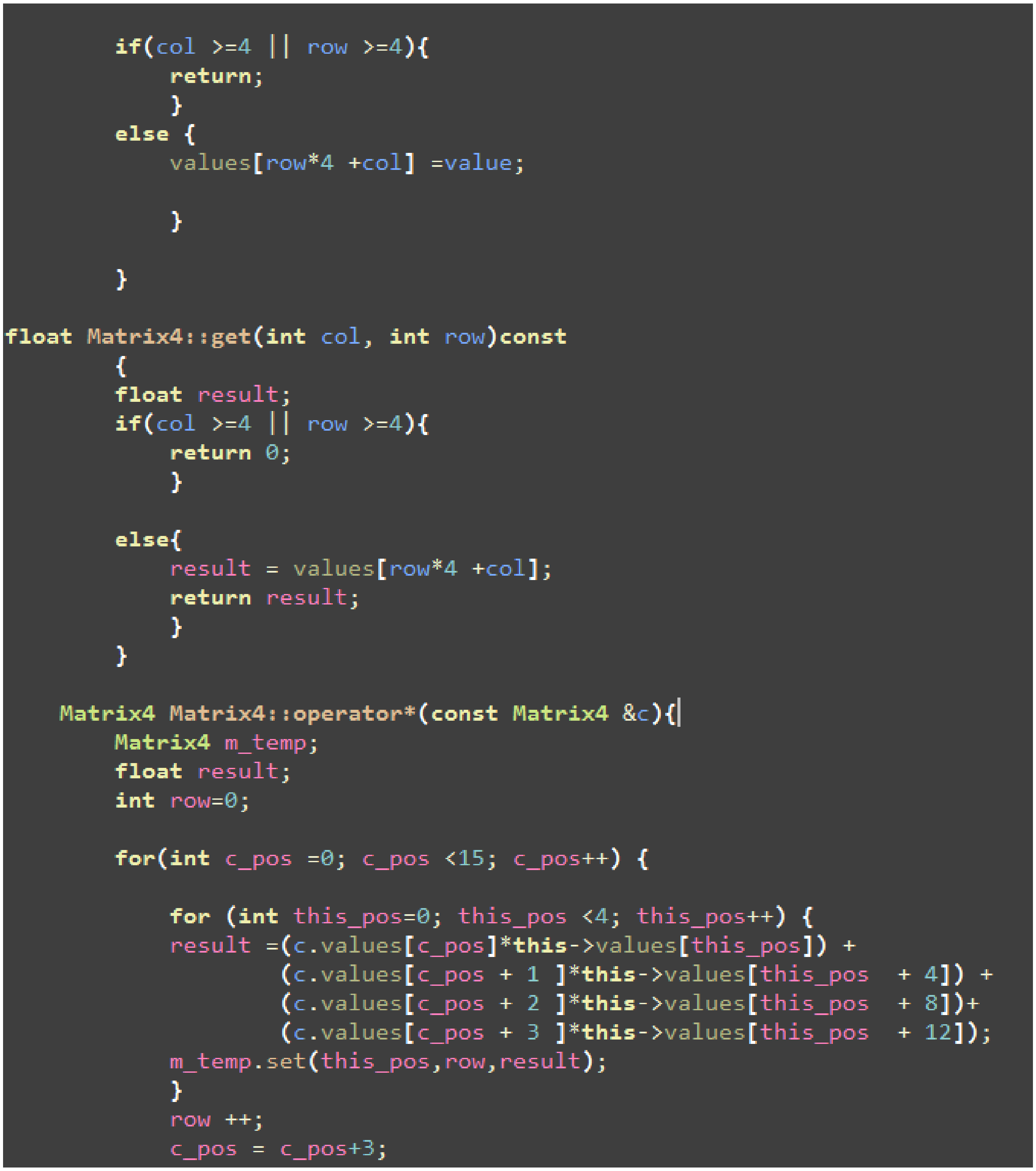
```
1000
0100
0010
0001

1300
0102
4010
0051

2
se libera el espacio
1300
0102
4010
0051

se libera el espacio
se libera el espacio
se libera el espacio
```



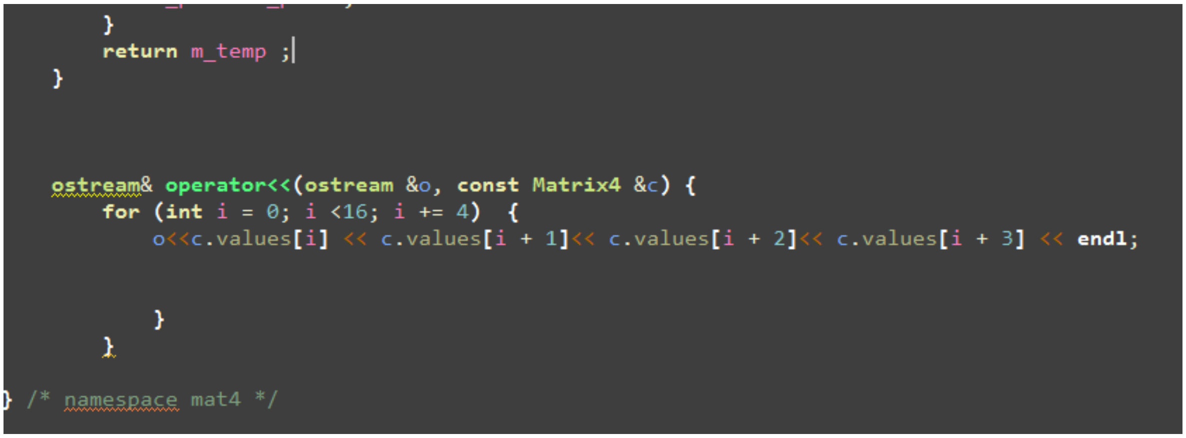
```
 * Matrix4.h

#ifndef MATRIX4_H_
#define MATRIX4_H_
#include <iostream>
using namespace std;

namespace mat4 {

class Matrix4 {

public:
    float values[16];
    Matrix4();
    virtual ~Matrix4();
    void setIdentity();
    void set(int col, int row, float value);
    float get(int col, int row)const;
    Matrix4 operator*(const Matrix4 &C);
    friend ostream& operator <<(ostream &o, const Matrix4 &c);
};

} /* namespace mat4 */

#endif /* MATRIX4_H_ */
```



```
 * Matrix4.cpp

#include "Matrix4.h"
#include <iostream>
#include <cmath>

namespace mat4 {
// Constructor
Matrix4::Matrix4() {
    // TODO Auto-generated constructor stub
    setIdentity();

}

Matrix4::~Matrix4() {
    using namespace std;
        cout << "se libera el espacio" << endl;
        // TODO Auto-generated destructor stub
    }

void Matrix4::setIdentity()
        {

    int position =0;
    int control = 5;

    for (position= 0; position < 16; position++){


        if (control ==5){
            values[position] = 1;
            control = 0;

            }
        else{
            values[position] = 0;
            }
        control++;
            }
        }
void Matrix4::set(int col, int row, float value){
```



```
        if(col >=4 || row >=4){
            return;
            }
        else {
            values[row*4 +col] =value;

            }

        }

float Matrix4::get(int col, int row)const
        {
        float result;
        if(col >=4 || row >=4){
            return 0;
            }

        else{
            result = values[row*4 +col];
            return result;
            }
        }

    Matrix4 Matrix4::operator*(const Matrix4 &c){
        Matrix4 m_temp;
        float result;
        int row=0;

        for(int c_pos =0; c_pos <15; c_pos++) {

            for (int this_pos=0; this_pos <4; this_pos++) {
            result =(c.values[c_pos]*this->values[this_pos]) +
                    (c.values[c_pos + 1 ]*this->values[this_pos  + 4]) +
                    (c.values[c_pos + 2 ]*this->values[this_pos  + 8])+
                    (c.values[c_pos + 3 ]*this->values[this_pos  + 12]);
            m_temp.set(this_pos,row,result);
            }
            row ++;
            c_pos = c_pos+3;
```



```
        }
        return m_temp ;
    }



    ostream& operator<<(ostream &o, const Matrix4 &c) {
        for (int i = 0; i <16; i += 4)  {
            o<<c.values[i] << c.values[i + 1]<< c.values[i + 2]<< c.values[i + 3] << endl;


            }
        }

} /* namespace mat4 */
```