```cpp
/**
  ******************************************************************************
  * @file    : main.cpp
  * @brief   : Main Program
  *          : Lab 2: Arrays and Matrices
  *          : CS-3210/021
  * @date    : MAR 23 2021
  * @author : Julian Singkham
  ******************************************************************************
  * @attention
  *
  *
  *
  ******************************************************************************
**/

#include "printm.h"
#include "multiplym.h"
#include <string>
#include <iostream>
#include <fstream>

using namespace std;

//=======================================Methods=========================================
 /**
  * @brief The program entry point
  *
  * @param NOT USED
  *
  * @retval NOT USED
  */
 int main(int argc, char **argv){
    // Test malformed matrix
    int rowsl = 2;
    int colsl = 2;
    double **matrix_l = new double*[rowsl];//Allocates memory to the # of rows

    //Allocate memory for each row of the array to the # of columns
    for(int i = 0; i < rowsl; i++)
        matrix_l[i] = new double[colsl];

    //Fill the 2D array with numbers, 0-4
    for(int i = 0; i < rowsl; i++)
        for(int j = 0; j < colsl; j++)
            matrix_l[i][j] = (j+1)+3*i;
    printm(matrix_l, rowsl, colsl);
    cout << endl;

    int rowsr = 3;
    int colsr = 2;
    double **matrix_r = new double*[rowsr]; //Allocates memory to the # of rows

    //Allocate memory for each row of the array to the amount of columns
    for(int i = 0; i < rowsr; i++)
        matrix_r[i] = new double[colsr];

    //Fill the 2D array with numbers, 5-10
    for(int i = 0; i < rowsr; i++)
        for(int j = 0; j < colsr; j++)
            matrix_r[i][j] = (j+7)+2*i;
    printm(matrix_r, rowsr, colsr);
    cout << endl;

    try{
        multiplym(matrix_l, rowsl, colsl, matrix_r, rowsr, colsr);
    }
    catch(matrix_Exception& e){
        cout << e.what() << endl;
    }

```

```
72      //Test proper matrix
73      for (int i = 0; i <  rowsl; i++)
74          delete[] matrix_l[i];
75      delete[] matrix_l;
76
77      rowsl = 2;
78      colsl = 3;
79      matrix_l = new double*[rowsl]; //Allocates memory to the # of rows
80
81      //Allocate memory for each row of the array to the amount of columns
82      for(int i = 0; i < rowsl; i++)
83          matrix_l[i] = new double[colsl];
84
85      for(int i = 0; i < rowsl; i++)
86          for(int j = 0; j < colsl; j++)
87              matrix_l[i][j] = (double) (j+1)+3*i;
88      printm(matrix_l, rowsl, colsl);
89      cout << endl;
90      printm(matrix_r, rowsr, colsr);
91      cout << endl;
92
93      try{
94          multiplym(matrix_l, rowsl, colsl, matrix_r, rowsr, colsr);
95      }
96      catch(matrix_Exception& e){
97          cout << e.what() << endl;
98      }
99
100     //Delete Arrays
101     for (int i = 0; i <  rowsl; i++)
102         delete[] matrix_l[i];
103     delete[] matrix_l;
104
105     for (int i = 0; i <  rowsr; i++)
106         delete[] matrix_r[i];
107     delete[] matrix_r;
108 }
```

```cpp
/**
  ******************************************************************************
  * @file    : matrixm.cpp
  * @brief   : matrix multiplier
  *          : Lab 2: Arrays and Matrices
  *          : CS-3210/021
  * @date    : MAR 23 2021
  * @author : Julian Singkham
  ******************************************************************************
  * @attention
  *
  *
  *
  ******************************************************************************
**/

#include "multiplym.h"
#include "printm.h"
#include <cfloat>
#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;

//=======================================Methods=====================================

// mxn * nxp = mxp
// inner values must equal
void multiplym(double ** lhs, int rowsl,int colsl, double ** rhs, int rowsr,int colsr){
    if (colsl != rowsr)
        throw matrix_Exception("Size mixmatch – The column of the left matrix does not"
                               "match therow of the right matrix:");

  double **matrix_m = new double*[rowsl];//Allocates memory to the # of rows

  //Allocate memory for each row of the array to the # of columns
  for(int i = 0; i < rowsl; i++)
      matrix_m[i] = new double[colsr];

    for (int i = 0; i < rowsl; ++i)
        for (int j = 0; j < colsr; ++j)
            for (int k = 0; k < colsl; ++k)
                matrix_m[i][j] += lhs[i][k] * rhs[k][j];

  printm(matrix_m, rowsl, colsr);

  for (int i = 0; i <  rowsl; i++)
        delete[] matrix_m[i];
  delete[] matrix_m;

    return;
}
```

```
/**
  ****************************************************************************
  * @file    : stlFileReader.h
  * @brief   : Outline for the matrix multiplier
  *          : Lab 2: Arrays and Matrices
  *          : CS-3210/021
  * @date    : MAR 23 2021
  * @author : Julian Singkham
  ****************************************************************************
**/
#ifndef MULTIPLYM
#define MULTIPLYM

#include <stdexcept>

using namespace std;


class matrix_Exception:public runtime_error{
    public:
        matrix_Exception(string message):
            runtime_error((string("A Matrix Error has occured: ") +
                message).c_str()) {}
};

void multiplym(double ** lhs, int rowsl,int colsl, double ** rhs, int rowsr,int colsr);

#endif
```

```cpp
/**
  ******************************************************************************
  * @file    : printm.cpp
  * @brief   : matrix printer
  *          : Lab 2: Arrays and Matrices
  *          : CS-3210/021
  * @date    : MAR 23 2021
  * @author : Julian Singkham
  ******************************************************************************
  * @attention
  *
  *
  *
  ******************************************************************************
**/

#include "printm.h"
#include <cfloat>
#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;

//=======================================Methods=======================================
/**
 * @brief Prints the contents of a matrix (2-D array)
 *
 * @param: **m: Pointer to a double matrix
 * @param: rows: How many rows the matrix has
 * @param: cols: How many columns the matrix has
 *
 * @retval NONE
 */
void printm(double **m, int rows, int cols){
    for(int i = 0; i < rows; i++){
        cout << "|";
        for(int j = 0; j < cols; j++){
            double temp = m[i][j];
            cout << temp << "|";
        }
        cout << endl;
    }
}
```

```
/**
  ********************************************************************************
  * @file    : printm.h
  * @brief   : Outline for matrix printer
  *          : Lab 2: Arrays and Matrices
  *          : CS-3210/021
  * @date    : MAR 13 2021
  * @author : JulianSingkham
  ********************************************************************************
**/

#ifndef PRINTM
#define PRINTM

using namespace std;

/**
 * @brief Prints the contents of a matrix (2-D array)
 *
 * @param: **m: Pointer to a double matrix
 * @param: rows: How many rows the matrix has
 * @param: cols: How many columns the matrix has
 *
 * @retval NONE
 */
void printm(double **m, int rows, int cols);

#endif
```

```
CC = g++
CFLAGS = -c -MMD
LFLAGS = -Wall -Wextra -g
LDFLAGS ?= -lglut -lGLU  -lGL
SOURCES = $(wildcard *.cpp)
OBJECTS = $(SOURCES:.cpp=.o)
EXECUTABLE = ex

all: $(EXECUTABLE) clean
$(EXECUTABLE): $(OBJECTS)
    $(CC) $(LFLAGS) -o $@ $(OBJECTS) $(LDFLAGS)

%.o:%.cpp
    $(CC) $(CFLAGS) $<

-include *.d

clean:
    rm -f *.d
    rm -f *.o
```

Table of Contents