

```

1  /**
2   * *****
3   * @file    : main.cpp
4   * @brief   : Main Program
5   *         : Lab 2: Arrays and Matrices
6   *         : CS-3210/021
7   * @date    : MAR 23 2021
8   * @author  : Julian Singkham
9   * *****
10  * @attention
11  *
12  *
13  *
14  * *****
15  */
16
17 #include "printm.h"
18 #include "multiplm.h"
19 #include <string>
20 #include <iostream>
21 #include <fstream>
22
23 using namespace std;
24
25 //=====Methods=====
26 /**
27  * @brief The program entry point
28  *
29  * @param NOT USED
30  *
31  * @retval NOT USED
32  */
33 int main(int argc, char **argv){
34     // Test malformed matrix
35     int rowsl = 2;
36     int colsl = 2;
37     double **matrix_l = new double*[rowsl]; //Allocates memory to the # of rows
38
39     //Allocate memory for each row of the array to the # of columns
40     for(int i = 0; i < rowsl; i++)
41         matrix_l[i] = new double[colsl];
42
43     //Fill the 2D array with numbers, 0-4
44     for(int i = 0; i < rowsl; i++)
45         for(int j = 0; j < colsl; j++)
46             matrix_l[i][j] = (j+1)+3*i;
47     printm(matrix_l, rowsl, colsl);
48     cout << endl;
49
50     int rowsr = 3;
51     int colsr = 2;
52     double **matrix_r = new double*[rowsr]; //Allocates memory to the # of rows
53
54     //Allocate memory for each row of the array to the amount of columns
55     for(int i = 0; i < rowsr; i++)
56         matrix_r[i] = new double[colsr];
57
58     //Fill the 2D array with numbers, 5-10
59     for(int i = 0; i < rowsr; i++)
60         for(int j = 0; j < colsr; j++)
61             matrix_r[i][j] = (j+7)+2*i;
62     printm(matrix_r, rowsr, colsr);
63     cout << endl;
64
65     try{
66         multiplm(matrix_l, rowsl, colsl, matrix_r, rowsr, colsr);
67     }
68     catch(matrix_Exception& e){
69         cout << e.what() << endl;
70     }
71

```

```
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     multiplm(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 }
```

```

1  /**
2   * *****
3   * @file    : matrixm.cpp
4   * @brief   : matrix multiplier
5   *         : Lab 2: Arrays and Matrices
6   *         : CS-3210/021
7   * @date    : MAR 23 2021
8   * @author  : Julian Singkham
9   * *****
10  * @attention
11  *
12  *
13  *
14  * *****
15  **/
16
17 #include "multiplm.h"
18 #include "printm.h"
19 #include <cmath>
20 #include <iostream>
21 #include <fstream>
22 #include <iomanip>
23
24 using namespace std;
25
26 //=====Methods=====
27
28 // mxn * nxp = mxp
29 // inner values must equal
30 void multiplm(double ** lhs, int rowsl, int colsl, double ** rhs, int rowsr, int colsr){
31     if (colsl != rowsr)
32         throw matrix_Exception("Size mismatch - The column of the left matrix does not"
33                                "match the row of the right matrix:");
34
35     double **matrix_m = new double*[rowsl]; //Allocates memory to the # of rows
36
37     //Allocate memory for each row of the array to the # of columns
38     for(int i = 0; i < rowsl; i++)
39         matrix_m[i] = new double[colsr];
40
41     for (int i = 0; i < rowsl; ++i)
42         for (int j = 0; j < colsr; ++j)
43             for (int k = 0; k < colsl; ++k)
44                 matrix_m[i][j] += lhs[i][k] * rhs[k][j];
45
46     printm(matrix_m, rowsl, colsr);
47
48     for (int i = 0; i < rowsl; i++)
49         delete[] matrix_m[i];
50     delete[] matrix_m;
51
52     return;
53 }

```

```
1  /**
2   * *****
3   * @file    : stlFileReader.h
4   * @brief   : Outline for the matrix multiplier
5   *          : Lab 2: Arrays and Matrices
6   *          : CS-3210/021
7   * @date    : MAR 23 2021
8   * @author  : Julian Singkham
9   * *****
10 */
11 #ifndef MULTIPLYM
12 #define MULTIPLYM
13
14 #include <stdexcept>
15
16 using namespace std;
17
18 class matrix_Exception:public runtime_error{
19 public:
20     matrix_Exception(string message):
21         runtime_error((string("A Matrix Error has occured: ") +
22             message).c_str()) {}
23 };
24
25 void multiplm(double ** lhs, int rowsl,int colsl, double ** rhs, int rowsr,int colsr);
26
27 #endif
28
```

```

1  /**
2      *****
3      * @file      : printm.cpp
4      * @brief     : matrix printer
5      *           : Lab 2: Arrays and Matrices
6      *           : CS-3210/021
7      * @date      : MAR 23 2021
8      * @author    : Julian Singkham
9      *****
10     * @attention
11     *
12     *
13     *
14     *****
15 **/
16
17 #include "printm.h"
18 #include <cstdio>
19 #include <iostream>
20 #include <fstream>
21 #include <iomanip>
22
23 using namespace std;
24
25 //=====Methods=====
26 /**
27  * @brief Prints the contents of a matrix (2-D array)
28  *
29  * @param: **m: Pointer to a double matrix
30  * @param: rows: How many rows the matrix has
31  * @param: cols: How many columns the matrix has
32  *
33  * @retval NONE
34  */
35 void printm(double **m, int rows, int cols){
36     for(int i = 0; i < rows; i++){
37         cout << "|";
38         for(int j = 0; j < cols; j++){
39             double temp = m[i][j];
40             cout << temp << "|";
41         }
42         cout << endl;
43     }
44 }

```

```
1  /**
2   * *****
3   * @file      : printm.h
4   * @brief     : Outline for matrix printer
5   *           : Lab 2: Arrays and Matrices
6   *           : CS-3210/021
7   * @date      : MAR 13 2021
8   * @author    : JulianSingkham
9   * *****
10 */
11
12 #ifndef PRINTM
13 #define PRINTM
14
15 using namespace std;
16
17 /**
18  * @brief Prints the contents of a matrix (2-D array)
19  *
20  * @param: **m: Pointer to a double matrix
21  * @param: rows: How many rows the matrix has
22  * @param: cols: How many columns the matrix has
23  *
24  * @retval NONE
25  */
26 void printm(double **m, int rows, int cols);
27
28 #endif
```

```
1 CC = g++
2 CFLAGS = -c -MMD
3 LFLAGS = -Wall -Wextra -g
4 LDFLAGS ?= -lglut -lGLU -lGL
5 SOURCES = $(wildcard *.cpp)
6 OBJECTS = $(SOURCES:.cpp=.o)
7 EXECUTABLE = ex
8
9 all: $(EXECUTABLE) clean
10 $(EXECUTABLE): $(OBJECTS)
11     $(CC) $(LFLAGS) -o $@ $(OBJECTS) $(LDFLAGS)
12
13 %.o: %.cpp
14     $(CC) $(CFLAGS) $<
15
16 -include *.d
17
18 clean:
19     rm -f *.d
20     rm -f *.o
```

1	Table of Contents								
2	1 main.cpp.....	sheets	1 to	2 (2)	pages	1-	2	109	lines
3	2 multiplm.cpp.....	sheets	3 to	3 (1)	pages	3-	3	54	lines
4	3 multiplm.h.....	sheets	4 to	4 (1)	pages	4-	4	29	lines
5	4 printm.cpp.....	sheets	5 to	5 (1)	pages	5-	5	45	lines
6	5 printm.h.....	sheets	6 to	6 (1)	pages	6-	6	29	lines
7	6 Makefile.....	sheets	7 to	7 (1)	pages	7-	7	21	lines