

# CS205 C/ C++ Program Design

## Assignment 4

Name: Kimhan Teng, SID: 11710526

### Part 1. Source Code and Description

Actually, in this assignment we divided into three function file. There are main.cpp, matrix.cpp and Matrix.hpp. Actually, a class Matrix has been designed for a matrix with float elements and it contain three private members attribute which are the number of rows and columns, data(p) with a float data type. It has a constructors, and some operation like << for print out, >> for input, \* operation overloading for multiplication such that the class can support:  $C = A * B$ ,  $C = A * b$ , and  $C = a * B$  (Scalar and Matrix) and we also implement + operation for addition.

#### 1) Matrix.hpp

```
↳ Matrix.hpp ×
↳ Matrix.hpp > ...
1  #ifndef MATRIX_H_INCLUDED
2  #define MATRIX_H_INCLUDED
3  #include <iostream>
4  #include<cassert>
5  #include <iomanip>
6  #include<cmath>
7  using namespace std;
8
9
10 class Matrix
11 {
12     friend ostream& operator<<(ostream &, const Matrix&);
13     friend istream& operator>>(istream &, Matrix&);
14     friend Matrix operator*(const Matrix&, const Matrix&);
15     friend Matrix operator*(const Matrix&, float num);
16     friend Matrix operator*(float num, const Matrix&);
17     private:
18         int rows, columns;
19         float **p;
20     public:
21         Matrix(int=1, int=1);
22         Matrix operator+(const Matrix&) const;
23 };
24 #endif // MATRIX_H_INCLUDED
```

## 2) matrix.cpp

```
matrix.cpp ●
G matrix.cpp > %+ operator>>(istream &, Matrix &)
1 #include "Matrix.hpp"
2
3 //for cout operation
4 ostream &operator<<( ostream &cout, const Matrix& n )
5 {
6     for(int i=0; i<n.rows; i++)
7     {
8         for(int j=0; j<n.columns; j++)
9         {
10             cout << setw(14) << n.p[i][j];
11         }
12         cout << endl;
13     }
14     return cout;
15 }
16
17 // for cin operation
18 istream &operator>>( istream &ciin, Matrix &n )
19 {
20     delete [] n.p;
21     cout<<"Enter the number of Rows and Columns: ";
22     ciin>>n.rows>>n.columns;
23     n.p = new float *[n.rows];
24     for(int i=0; i<n.rows; i++) [
25         n.p[i] = new float[n.columns];
26     ]
27     cout << "Enter the " << n.rows << " rows of the matrix:" << endl;
28
29     for(int i=0; i<n.rows; i++)
30     {
31         cout << "Enter the " << n.columns << " elements of row number " << i+1 << ": ";
32         for(int j=0; j<n.columns; j++)
33         {
34             ciin >> n.p[i][j];
35         }
36     }
37     return ciin;
38 }
```

```

40 // for additional operation
41 Matrix Matrix::operator+(const Matrix& n) const
42 {
43     int i, j;
44     Matrix tempMatrix(n.rows, n.columns);
45
46     for(i=0; i<n.rows; i++)
47     {
48         for(j=0; j<n.columns; j++)
49         {
50             tempMatrix.p[i][j] = p[i][j] + n.p[i][j];
51         }
52     }
53     return tempMatrix;
54 };
55
56 // multiple operation
57 Matrix operator*(const Matrix& A, const Matrix& B)
58 {
59     float sum;
60     int i, j, k, n, m, l;
61     n = A.rows;
62     m = A.columns;
63     l = B.columns;
64     Matrix tempMatrix(n, l);
65
66     for(i=0; i<n; i++)
67     {
68         for(j=0; j<l; j++)
69         {
70             sum = 0;
71             for(k=0; k<m ; k++)
72             {
73                 sum = sum + A.p[i][k] * B.p[k][j];
74             }
75             tempMatrix.p[i][j] = sum;
76         }
77     }
78     return tempMatrix;
79 }
80 // right mult
81 Matrix operator*(const Matrix& A, float num)
82 {
83     Matrix tempMatrix(A.rows, A.columns);
84
85     for(int i=0;i<A.rows;i++)
86     {
87         for(int j=0;j<A.columns;j++)
88         {
89             tempMatrix.p[i][j] = A.p[i][j] * num;
90         }
91     }
92     return tempMatrix;
93 }
94
95 //left mult
96 Matrix operator*(float num, const Matrix& B)
97 {
98     Matrix tempMatrix(B.rows, B.columns);
99     for(int i=0;i<B.rows;i++)
100    {
101        for(int j=0;j<B.columns;j++)
102        {
103            tempMatrix.p[i][j] = B.p[i][j] * num;
104        }
105    }
106    return tempMatrix;
107 }
108 }
```

```
110     Matrix::Matrix(int r, int c)
111     {
112         int i, j;
113         rows = r;
114         columns = c;
115
116         p = new float *[rows];
117         assert(p != NULL);
118         for(i=0; i<rows; i++)
119         {
120             p[i] = new float[columns];
121             assert(p[i] != NULL);
122         }
123
124         for(i=0; i<rows; i++)
125         {
126             for(j=0; j<columns; j++)
127             {
128                 p[i][j] = 0;
129             }
130         }
131     }
```

### 3) main.cpp

```
⌚ main.cpp ×
⌚ main.cpp > ⚡ main()
1 #include "Matrix.hpp"
2 #include <iostream>
3 using namespace std;
4
5 int main() []
6
7     Matrix A, B;
8     cout << "Enter the A matrix" << endl;
9     cin >> A;
10    cout << "Enter the B matrix" << endl;
11    cin >> B;
12
13    cout << "(A * b): A * 2" << endl;
14    cout << A * 2 << endl;
15
16    cout << "(a * B): 3 * B" << endl;
17    cout << 3 * B << endl;
18
19    cout << "The two matrices multiplication" << endl;
20    cout << A * B << endl;
21
22    cout << "The two matrices addition" << endl;
23    cout << A + B << endl;
24
25    return 0;
26 ]
```

## Part 2. Result & Verification

Here we have compile and run the program on laptop and ARM board. To run the program in ARM board with efficient way we have used cmake to manage with it.

### Test case #1:

The result here has been compile and run in vscode on Mac. Basically, the result below describe the matrix multiplication, scalar and also addition matrix.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
(base) Tengs-MacBook-Pro:Assignment3 tengkimhan$ ./a.out
Enter the A matrix
Enter the number of Rows and Columns: 4 4
Enter the 4 rows of the matrix:
Enter the 4 elements of row number 1: 1 1 1 1
Enter the 4 elements of row number 2: 1 1 1 1
Enter the 4 elements of row number 3: 1 1 1 1
Enter the 4 elements of row number 4: 1 1 1 1
Enter the B matrix
Enter the number of Rows and Columns: 4 4
Enter the 4 rows of the matrix:
Enter the 4 elements of row number 1: 1 2 2 2
Enter the 4 elements of row number 2: 1 2 2 2
Enter the 4 elements of row number 3: 1 2 2 2
Enter the 4 elements of row number 4: 2 2 2 2
(A * b): A * 2
      2          2          2          2
      2          2          2          2
      2          2          2          2
      2          2          2          2

(a * B): 3 * B
      3          6          6          6
      3          6          6          6
      3          6          6          6
      6          6          6          6

The two matrices multiplication
      5          8          8          8
      5          8          8          8
      5          8          8          8
      5          8          8          8

The two matrices addition
      2          3          3          3
      2          3          3          3
      2          3          3          3
      3          3          3          3
```

### Test case #2:

Here we have been compile and run the program on ARM board and used cmake to manage with this program.

- compile and run with cmake

We got the result the same as above.

```
[openailab@localhost ~]$ cd Desktop
[openailab@localhost Desktop]$ cd Assignment4/
[openailab@localhost Assignment4]$ cmake .
-- The C compiler identification is GNU 8.1.1
-- The CXX compiler identification is GNU 8.1.1
-- Check for working C compiler: /bin/cc
-- Check for working C compiler: /bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /bin/c++
-- Check for working CXX compiler: /bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/openailab/Desktop/Assignment4
[openailab@localhost Assignment4]$ make
-- Configuring done
-- Generating done
-- Build files have been written to: /home/openailab/Desktop/Assignment4
Scanning dependencies of target Assignment4
make[2]: Warning: File 'main.cpp' has modification time 81249133 s in the future
[ 33%] Building CXX object CMakeFiles/Assignment4.dir/main.cpp.o
[ 66%] Building CXX object CMakeFiles/Assignment4.dir/matrix.cpp.o
[100%] Linking CXX executable Assignment4
make[2]: warning: Clock skew detected. Your build may be incomplete.
[100%] Built target Assignment4
[openailab@localhost Assignment4]$ ./Assignment4
Enter the A matrix
Enter the number of Rows and Columns: 4 4
Enter the 4 rows of the matrix:
Enter the 4 elements of row number 1: 1 1 1 1
Enter the 4 elements of row number 2: 1 1 1 1
Enter the 4 elements of row number 3: 1 1 1 1
Enter the 4 elements of row number 4: 1 1 1 1
Enter the B matrix
Enter the number of Rows and Columns: 4 4
Enter the 4 rows of the matrix:
Enter the 4 elements of row number 1: 1 2 2 2
Enter the 4 elements of row number 2: 1 2 2 2
Enter the 4 elements of row number 3: 1 2 2 2
Enter the 4 elements of row number 4: 2 2 2 2
(A * b): A * 2
      2      2      2      2
      2      2      2      2
      2      2      2      2
      2      2      2      2
(a * B): 3 * B
      3      6      6      6
      3      6      6      6
      3      6      6      6
      6      6      6      6
The two matrices multiplication
      5      8      8      8
      5      8      8      8
      5      8      8      8
      5      8      8      8
The two matrices addition
      2      3      3      3
      2      3      3      3
      2      3      3      3
      3      3      3      3
[openailab@localhost Assignment4]$
```

### **Part 3. Difficulties & Solutions, or others**

Finally, there are some problem and difficult that I have met, especially with the operation loading since it took me some times to understand. Anyway, I did the research on the internet and found solution.