Session 8 / Part 2

Operator Overloading (w/ Friend keyword)

Operator Overloading

Operator Overloading in C++ are: +, -, *, /, %, =, ==, !, etc. (You cannot overload all the operators)

Operator Overloading:

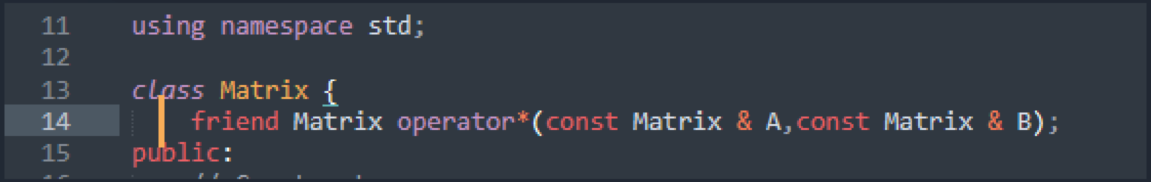
1. **Concept:**
    1. The same concept as Function Overloading. It has different syntax, however.

2. **Necessity:**
    1. Obviously, it makes your program easier to read and work with.
    2. Makes it possible for templates to work equally well with classes **.**

3. **Syntax:** There is a lot here to over, but shortly, we have two types of Operator overloading:
    1. Overloading as an instance member functions
    2. **Overloading as a friend function**

Lab Exercise:

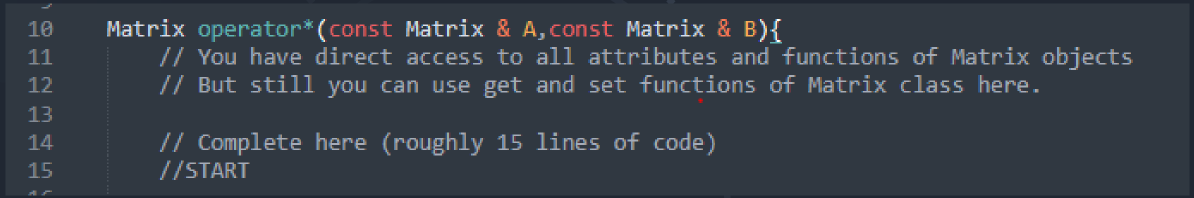
1. You already know "overloading" concept & "Friend" concept. Overloading Operators as a friend function is a combination of these two.
2. Operator* is already declared as a friend in Matrix class



```
using namespace std;

class Matrix {
    friend Matrix operator*(const Matrix & A,const Matrix & B);
public:
```

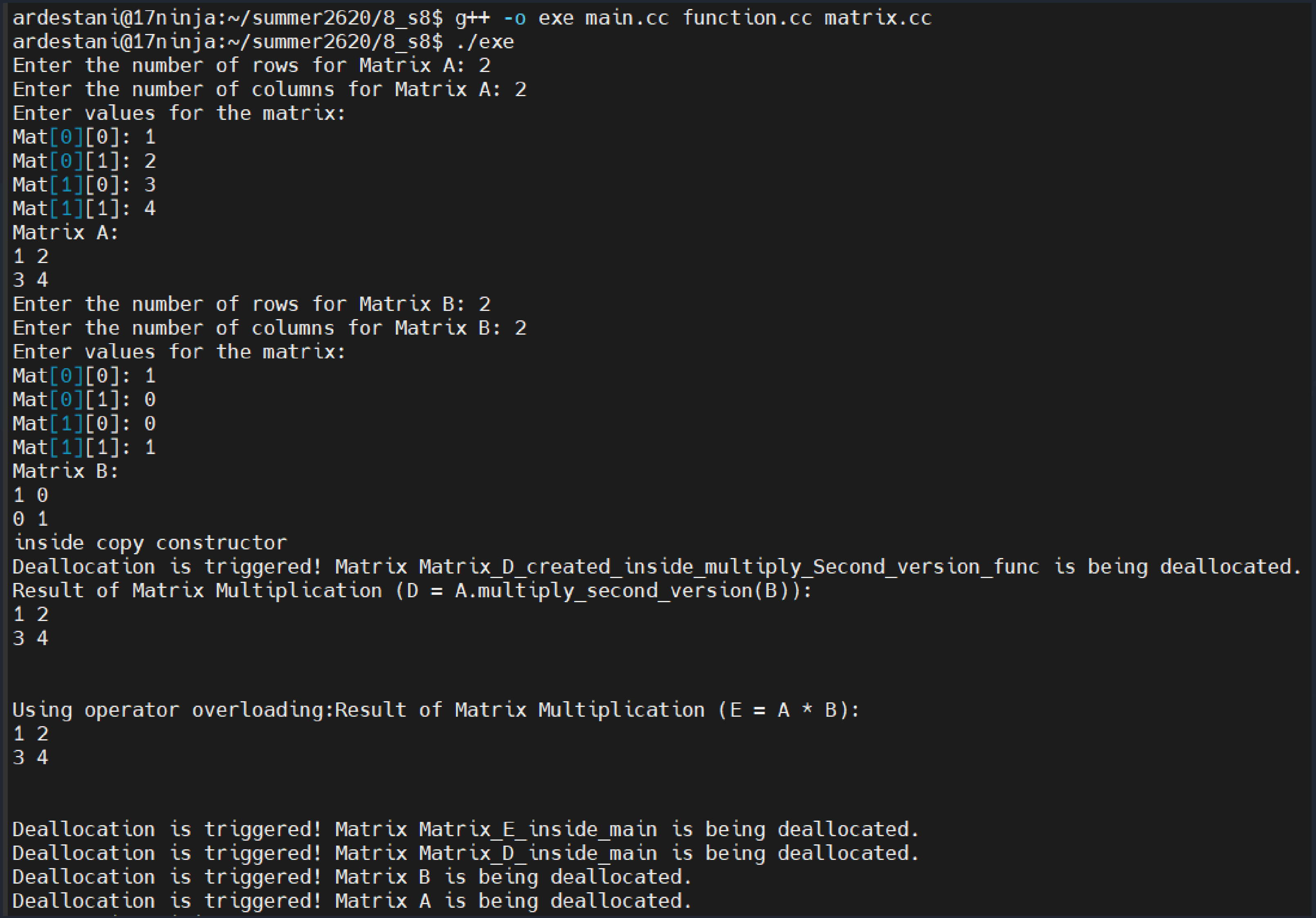
3. Open function.cc file and implement this function.



```
Matrix operator*(const Matrix & A,const Matrix & B){
    // You have direct access to all attributes and functions of Matrix objects
    // But still you can use get and set functions of Matrix class here.

    // Complete here (roughly 15 lines of code)
    //START
```

Expected output:



```
ardestani@17ninja:~/summer2620/8_s8$ g++ -o exe main.cc function.cc matrix.cc
ardestani@17ninja:~/summer2620/8_s8$ ./exe
Enter the number of rows for Matrix A: 2
Enter the number of columns for Matrix A: 2
Enter values for the matrix:
Mat[0][0]: 1
Mat[0][1]: 2
Mat[1][0]: 3
Mat[1][1]: 4
Matrix A:
1 2
3 4
Enter the number of rows for Matrix B: 2
Enter the number of columns for Matrix B: 2
Enter values for the matrix:
Mat[0][0]: 1
Mat[0][1]: 0
Mat[1][0]: 0
Mat[1][1]: 1
Matrix B:
1 0
0 1
inside copy constructor
Deallocation is triggered! Matrix Matrix_D_created_inside_multiply_Second_version_func is being deallocated.
Result of Matrix Multiplication (D = A.multiply_second_version(B)):
1 2
3 4


Using operator overloading:Result of Matrix Multiplication (E = A * B):
1 2
3 4


Deallocation is triggered! Matrix Matrix_E_inside_main is being deallocated.
Deallocation is triggered! Matrix Matrix_D_inside_main is being deallocated.
Deallocation is triggered! Matrix B is being deallocated.
Deallocation is triggered! Matrix A is being deallocated.
```

We will discuss Operator Overloading more at session #10