

Assignment No.: 6

● Problem Statement:

Write a Program in C++ to create a class LOG having two data members x and y to represent $x \log_b y$ where the base b is a constant given by the user. Perform four basic operations: addition, subtraction, power, root using operator overloading.

● Algorithm:

→ **Name of the class :-** Log

→ **Private data members of the class :-** x, y

→ **Public member functions of the class :**

- ✓ operator+()
- ✓ operator-()
- ✓ power()
- ✓ Log():x(0),y(0) //Constructor

→ **Algorithm for method operator +():**

- Step 1. Set p1 = pow(y,x)
- Step 2. Set p2 = pow(b.y , b.x)
- Step 3. Call Log n(1, p1*p2)
- Step 4. Return n

→ **Algorithm for method operator -():**

- Step 1. Set p1 = pow(y,x)
- Step 2. Set p2 = pow(b.y , b.x)
- Step 3. Call Log n(1, p1/p2)
- Step 4. Return n

→ **Algorithm for method power():**

- Step 1. Set p1 = pow (y,x)
- Step 2. Call Log n(1,p1)
- Step 3. Return n

→ **Algorithm for method main():**

- Step 1. Print "Enter x and y for the first term in the form (x log y)";
- Step 2. Read x,y
- Step 3. Set l1 = Log(x,y)
- Step 4. Print "Enter x and y for the second term in the form (x log y)";
- Step 5. Read x,y
- Step 6. Set l2 = Log(x,y)
- Step 7. Print "l1 + l2"
- Step 8. Set Log l3 = l1 + l2
- Step 9. Print "l3"

Date:

Step 10. Set $I3 = I1 - I2$
Step 11. Print " $I1 - I2$ "
Step 12. Print "Applying power rule :"
Step 13. Set $I3 = I1.power()$
Step 14. Print " $I1 \rightarrow I3$ "
Step 15. Set $I3 = I2.power()$
Step 16. Print " $I2 \rightarrow I3$ "

● **Source Code:**

```
#include <iostream>
#include <math.h>

using namespace std;

class Log {
private:
    double x, y;    // x log y
public:
    Log() : x(0), y(0) {}
    Log(double _x, double _y) : x(_x), y(_y) {}

    Log operator+(Log &b) {
        double p1 = pow(y, x);
        double p2 = pow(b.y, b.x);
        Log n(1, p1*p2);
        return n;
    }

    Log operator-(Log &b) {
        double p1 = pow(y, x);
        double p2 = pow(b.y, b.x);
        Log n(1, p1/p2);
        return n;
    }

    Log power() {
        double p1 = pow(y, x);
        Log n(1, p1);
        return n;
    }

    friend ostream& operator<<(ostream& os, Log &l);
};

ostream& operator<<(ostream& os, Log &l) {
    return os << l.x << " log " << l.y;
```

Date:

```
}  
  
int main() {  
    double x, y;  
    Log l1, l2;  
    cout << "Enter x and y for the first term in the form (x log y)" << endl;  
    cout << "Enter x : ";  
    cin >> x;  
    cout << "Enter y : ";  
    cin >> y;  
    l1 = Log(x, y);  
    cout << "Enter x and y for the second term in the form (x log y)" << endl;  
    cout << "Enter x : ";  
    cin >> x;  
    cout << "Enter y : ";  
    cin >> y;  
    l2 = Log(x, y);  
    cout << l1 << " + " << l2 << " : ";  
    Log l3 = l1 + l2;  
    cout << l3 << endl;  
    l3 = l1 - l2;  
    cout << l1 << " - " << l2 << " : " << l3 << endl;  
    cout << "Applying power rule : " << endl;  
    l3 = l1.power();  
    cout << l1 << " -> " << l3 << endl;  
    l3 = l2.power();  
    cout << l2 << " -> " << l3 << endl;  
}
```

● Input & Output:

Enter x and y for the first term in the form (x log y)

Enter x : 5

Enter y : 6

Enter x and y for the second term in the form (x log y)

Enter x : 2

Enter y : 4

5 log 6 + 2 log 4 : 1 log 124416

5 log 6 - 2 log 4 : 1 log 486

Applying power rule :

Date:

5 log 6 -> 1 log 7776

2 log 4 -> 1 log 16

- **Discussion:**

1. Name of an operator function is always operator keyword followed by symbol of operator and operator functions are called when the corresponding operator is used.
2. For operator overloading to work, at least one of the operands must be a user defined class object.
3. We can also write conversion operators that can be used to convert one type to another type.