



☆ Overloading + operator



MinMax class find the minimum and maximum values from the given values.

MinMax class has following data members and public member functions

int m_nMin ,int m_nMax

MinMax() : Default constructor to initialize data members of class with zero.

MinMax(int,int): Parameterized constructor

int GetMin() ; // it will return m_nMin

int GetMax() //it will return m_nMax

void getdata() //Function to enter values from keyboard.

friend MinMax operator+(const MinMax &cM1, const MinMax &cM2); //Will find max and min value in object cM1 and cM2

friend MinMax operator+(const MinMax &cM, int nValue); // will find max and min value in object cM and variable nValue

friend MinMax operator+(int nValue, const MinMax &cM); //will find max and min value between nValue variable and object cM

Sample Input

10 15

8 11

3 12

Sample output

Result:(3, 16)

3 is minimum number is out of 10 ,15,8 ,11,3 ,12,5,8,16

16 is maximum number is out of 10,15,8,11,3,12,5,8,16

YOUR ANSWER

[Original code](#)

C++



```
1 ▶ #include<iostream>
2 using namespace std;
3 class MinMax
4 {
5 private:
6     int m_nMin; // The min value
7     int m_nMax; // The max value
8
9 public:
10    MinMax();
11    MinMax(int nMin, int nMax);
12    int GetMin() { return m_nMin; }
13    int GetMax() { return m_nMax; }
14    void getdata();
15    friend MinMax operator+(const MinMax &cM1, const MinMax &cM2);
16    friend MinMax operator+(const MinMax &cM, int nValue);
17    friend MinMax operator+(int nValue, const MinMax &cM);
18 };
19
20 MinMax::MinMax()
21 {
22     m_nMin=0;
23     m_nMax=0;
24 }
25 MinMax::MinMax(int nMin,int nMax)
26 {
27     m_nMin=nMin;
28     m_nMax=nMax;
29 }
30 void MinMax::getdata()
31 {
32     cin>>m_nMin>>m_nMax;
33 }
34 MinMax operator+(const MinMax &cM1, const MinMax &cM2)
35 {
36     MinMax d;
37     d.m_nMin = cM1.m_nMin < cM2.m_nMin ? cM1.m_nMin : cM2.m_nMin;
38     d.m_nMax = cM1.m_nMax > cM2.m_nMax ? cM1.m_nMax : cM2.m_nMax;
39     return d;
40 }
41 MinMax operator+(const MinMax &cM, int nValue)
42 {
43     MinMax d;
44     d.m_nMin = cM.m_nMin < nValue ? cM.m_nMin : nValue;
45     d.m_nMax = cM.m_nMax > nValue ? cM.m_nMax : nValue;
46     return d;
47 }
48 MinMax operator+(int nValue, const MinMax &cM)
49 {
50     MinMax d;
51     d.m_nMin = nValue < cM.m_nMin ? nValue : cM.m_nMin;
52     d.m_nMax = nValue > cM.m_nMax ? nValue : cM.m_nMax;
53     return d;
54 }
```



```

39     }
40     else
41     {
42         d.m_nMin=cM2.m_nMin;
43     }
44     if(cM1.m_nMax>cM2.m_nMax)
45     {
46         d.m_nMax=cM1.m_nMax;
47     }
48     else
49     {
50         d.m_nMax=cM2.m_nMax;
51     }
52     return d;
53 }
54 MinMax operator+(const MinMax &cM,int nValue)
55 {
56     MinMax s;
57     if(cM.m_nMin<nValue)
58     {
59         s.m_nMin=cM.m_nMin;
60     }
61     else
62     {
63         s.m_nMin=nValue;
64     }
65     if(cM.m_nMax>nValue)
66     {
67         s.m_nMax=cM.m_nMax;
68     }
69     else
70     {
71         s.m_nMax=nValue;
72     }
73     return s;
74 }
75 MinMax operator+(int nValue,const MinMax &cM)
76 {
77     MinMax l;
78     if(cM.m_nMin<nValue)
79     {
80         l.m_nMin=cM.m_nMin;
81     }
82     else
83     {
84         l.m_nMin=nValue;
85     }
86     if(cM.m_nMax>nValue)
87     {
88         l.m_nMax=cM.m_nMax;
89     }
90     else
91     {
92         l.m_nMax=nValue;
93     }
94     return l;
95 }
96 int main()
97 {
98     MinMax cM1,cM2,cM3;
99     cM1.getdata();
100    cM2.getdata();
101    cM3.getdata();
102
103    MinMax cMFinal=cM1+cM2+5+8+cM3+16;
104
105    cout<<"Result:("&<<cMFinal.GetMin()<<" , "<<cMFinal.GetMax()<<" "<<endl;
106
107    return 0;

```

☐ Test against custom input

Run Code

Submit code & Continue

(You can submit any number of times)

[📄 Download sample test cases](#) *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

1

2

3

4

5

6

7

8

9

10

[About](#) [Privacy Policy](#) [Terms of Service](#)