

Open array.cpp Save ~ /C++Lab/asn10

```
1 #include <iostream>
2 using namespace std;
3
4 template <class T>
5 class Array
6 {
7     private:
8         T *ele;
9         int size;
10        T min;
11        T max;
12    public:
13        Array()
14        {
15            cout << "Enter the size of the array : " ;
16            cin >> size;
17            ele = new T[size];
18        }
19        Array(int s) : size(s) { ele = new T[size]; }
20        ~Array(){ delete[] ele; }
21        void getData();
22        Array operator+(Array&);
23        void calMinMax();
24        void display();
25 };
26
27 template <class T>
28 void Array<T>::getData()
29 {
30     cout << "Enter the array elements : " ;
31     for(int i = 0; i < size; ++i)
32         cin >> ele[i];
33 }
34
35 template <class T>
36 Array<T> Array<T>::operator+(Array<T> &a2) //***** & is imp*****
37 {
38     if(size != a2.size)
39     {
40         cout << "Array sizes are not equal" << endl;
41         exit(0);
42     }
43     Array<T> a3(size);
44     for(int i = 0; i < size; ++i)
45         a3.ele[i] = ele[i] + a2.ele[i];
46     return a3;
47 }
```

C++ Tab Width: 4 Ln 1, Col 1 INS

Open array.cpp Save ~ /C++Lab/asgn10

```
47 }
48
49 template <class T>
50 void Array<T>::calMinMax()
51 {
52     min = ele[0];
53     max = ele[0];
54     for(int i = 1; i < size; ++i)
55     {
56         if(ele[i] > max)
57             max = ele[i];
58         if(ele[i] < min)
59             min = ele[i];
60     }
61 }
62
63 template <class T>
64 void Array<T>::display()
65 {
66     cout << "The array elements are : ";
67     for(int i = 0; i < size; ++i)
68         cout << ele[i] << " ";
69     cout << endl << "The maximum element of the array is : " << max << endl;
70     cout << "The minimum element of the array is : " << min << endl << endl;
71 }
72
73 int main()
74 {
75     cout << "Integer : " << endl;
76     cout << "Array 1 : " << endl;
77     Array<int> a1;
78     a1.getData();
79     a1.calMinMax();
80     a1.display();
81     cout << "Array 2 : " << endl;
82     Array<int> a2;
83     a2.getData();
84     a2.calMinMax();
85     a2.display();
86     Array<int> a3 = a1 + a2;
87     cout << "Array 1 + Array 2 = " << endl;
88     a3.calMinMax();
89     a3.display();
90     cout << "Float : " << endl;
91     Array<float> f1;
92     cout << "Array 1 : " << endl;
93     f1.getData();
```

Bracket match found on line: 37 C++ Tab Width: 4 Ln 47, Col 2 INS

Open array.cpp Save

```
60     }
61 }
62
63 template <class T>
64 void Array<T>::display()
65 {
66     cout << "The array elements are : ";
67     for(int i = 0; i < size; ++i)
68         cout << ele[i] << " ";
69     cout << endl << "The maximum element of the array is : " << max << endl;
70     cout << "The minimum element of the array is : " << min << endl << endl;
71 }
72
73 int main()
74 {
75     cout << "Integer : " << endl;
76     cout << "Array 1 : " << endl;
77     Array<int> a1;
78     a1.getData();
79     a1.calMinMax();
80     a1.display();
81     cout << "Array 2 : " << endl;
82     Array<int> a2;
83     a2.getData();
84     a2.calMinMax();
85     a2.display();
86     Array<int> a3 = a1 + a2;
87     cout << "Array 1 + Array 2 = " << endl;
88     a3.calMinMax();
89     a3.display();
90     cout << "Float : " << endl;
91     Array<float> f1;
92     cout << "Array 1 : " << endl;
93     f1.getData();
94     f1.calMinMax();
95     f1.display();
96     Array<float> f2;
97     cout << "Array 2 : " << endl;
98     f2.getData();
99     f2.calMinMax();
100    f2.display();
101    Array<float> f3 = f1 + f2;
102    cout << "Array 1 + Array 2 = " << endl;
103    f3.calMinMax();
104    f3.display();
105    return 0;
106 }
```

Bracket match found on line: 74 C++ Tab Width: 4 Ln 106, Col 2 INS

```
deven@deven-VirtualBox: ~/C++Lab/asn10
deven@deven-VirtualBox:~/C++Lab/asn10$ g++ array.cpp
deven@deven-VirtualBox:~/C++Lab/asn10$ ./a.out
Integer :
Array 1 :
Enter the size of the array : 7
Enter the array elements : 5 2 4 8 7 9 1
The array elements are : 5 2 4 8 7 9 1
The maximum element of the array is : 9
The minimum element of the array is : 1

Array 2 :
Enter the size of the array : 7
Enter the array elements : 1 2 3 4 5 6 7
The array elements are : 1 2 3 4 5 6 7
The maximum element of the array is : 7
The minimum element of the array is : 1






Array 1 + Array 2 =
The array elements are : 6 4 7 12 12 15 8
The maximum element of the array is : 15
The minimum element of the array is : 4

Float :
Enter the size of the array : 8
Array 1 :
Enter the array elements : 2.5 4.5 7.6 8.2 1.1 2.3 9.8 9.9
The array elements are : 2.5 4.5 7.6 8.2 1.1 2.3 9.8 9.9
The maximum element of the array is : 9.9
The minimum element of the array is : 1.1

Enter the size of the array : 8
Array 2 :
Enter the array elements : 1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8
The array elements are : 1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8
The maximum element of the array is : 8.8
The minimum element of the array is : 1.1

Array 1 + Array 2 =
The array elements are : 3.6 6.7 10.9 12.6 6.6 8.9 17.5 18.7
The maximum element of the array is : 18.7
The minimum element of the array is : 3.6

deven@deven-VirtualBox:~/C++Lab/asn10$
```


Open ▾  bubbleSort.cpp
~/C++Lab/asgn10 Save    

```
1 #include <iostream>
2 using namespace std;
3
4 template <class T>
5 inline void Swap(T &x, T &y)
6 {
7     T temp = x;
8     x = y;
9     y = temp;
10 }
11
12 template <class T>
13 void bubbleSort(T arr[], int size)
14 {
15     for(int i = 0; i < size-1; ++i)
16         for(int j = 0; j < size-i-1; ++j)
17             if(arr[j] > arr[j+1])
18                 Swap<T>(arr[j], arr[j+1]); //Swap *****capital S in Swap, coz swap already exists
19 }
20
21 template <class T>
22 void printArray(T arr[], int size)
23 {
24     cout << "The sorted array elements are : " ;
25     for(int i = 0; i < size; ++i)
26         cout << arr[i] << " " ;
27     cout << endl << endl;
28 }
29
30 int main()
31 {
32     int n;
33     cout << "Enter the size of the integer array : " ;
34     cin >> n;
35     int a[n];
36     cout << "Enter the integer array elements : " ;
37     for(int i = 0; i < n; ++i)
38         cin >> a[i];
39     bubbleSort<int>(a, n);
40     printArray<int>(a, n);
41     int n2;
42     cout << "Enter the size of the char array : " ;
43     cin >> n2;
44     char ch[n2];
45     cout << "Enter the char array elements : " ;
46     for(int i = 0; i < n2; ++i)
47         cin >> ch[i];
```

C++ ▾ Tab Width: 4 ▾ Ln 11, Col 1 ▾ INS

Open

bubbleSort.cpp
~/C++Lab/asgn10

Save

```
5 inline void Swap(T &x, T &y)
6 {
7     T temp = x;
8     x = y;
9     y = temp;
10 }
11
12 template <class T>
13 void bubbleSort(T arr[], int size)
14 {
15     for(int i = 0; i < size-1; ++i)
16         for(int j = 0; j < size-i-1; ++j)
17             if(arr[j] > arr[j+1])
18                 Swap<T>(arr[j], arr[j+1]); //Swap *****capital S in Swap, coz swap already exists
19 }
20
21 template <class T>
22 void printArray(T arr[], int size)
23 {
24     cout << "The sorted array elements are : " ;
25     for(int i = 0; i < size; ++i)
26         cout << arr[i] << " " ;
27     cout << endl << endl;
28 }
29
30 int main()
31 {
32     int n;
33     cout << "Enter the size of the integer array : " ;
34     cin >> n;
35     int a[n];
36     cout << "Enter the integer array elements : " ;
37     for(int i = 0; i < n; ++i)
38         cin >> a[i];
39     bubbleSort<int>(a, n);
40     printArray<int>(a, n);
41     int n2;
42     cout << "Enter the size of the char array : " ;
43     cin >> n2;
44     char ch[n2];
45     cout << "Enter the char array elements : " ;
46     for(int i = 0; i < n2; ++i)
47         cin >> ch[i];
48     bubbleSort<char>(ch, n2);
49     printArray<char>(ch, n2);
50     return 0;
51 }
```

Bracket match found on line: 31

C++ Tab Width: 4 Ln 51, Col 2 INS

```
deven@deven-VirtualBox: ~/C++Lab/asn10
deven@deven-VirtualBox: ~/C++Lab/asn10$ g++ bubbleSort.cpp
deven@deven-VirtualBox: ~/C++Lab/asn10$ ./a.out
Enter the size of the integer array : 10
Enter the integer array elements : 7 2 5 3 6 8 5 1 2 7
The sorted array elements are : 1 2 2 3 5 5 6 7 7 8

Enter the size of the char array : 12
Enter the char array elements : a c h s e q z j n m e a
The sorted array elements are : a a c e e h j m n q s z

deven@deven-VirtualBox:~/C++Lab/asn10$
```

Open expression.cpp Save ~ /C++Lab/asn10

```
1 #include <iostream>
2 #include <cstring>
3 #include <cctype>
4 #include <cmath>
5 using namespace std;
6
7 template <class T>
8 class Stack
9 {
10     private:
11         const int max;
12         T node[50];
13         int top;
14     public:
15         Stack() : max(50), top(-1) {}
16         int isEmpty() { return (top == -1); }
17         T Top() { return ((top == -1) ? -1 : node[top]); }
18         void push(T ch);
19         T pop();
20 };
21
22 template <class T>
23 void Stack<T>::push(T ch)
24 {
25     if(top == max-1)
26     {
27         cout << "stack full" << endl;
28         exit(0);
29     }
30     node[++top] = ch;
31 }
32
33 template <class T>
34 T Stack<T>::pop()
35 {
36     if(top == -1)
37     {
38         cout << "stack empty" << endl;
39         exit(0);
40     }
41     return node[top--];
42 }
43
44 class Expression
45 {
46     private:
47         string infix;
```

C++ Tab Width: 4 Ln 1, Col 1 INS

Open expression.cpp Save

~/C++Lab/asn10

```
43
44 class Expression
45 {
46     private:
47         string infix;
48         double result;
49         int isOperator(char);
50         int isRightAssociative(char);
51         int getPrecedenceValue(char);
52         int hasGreaterPrecedence(char, char);
53         int isOpeningBracket(char);
54         int isClosingBracket(char);
55         double performOperation(char, double, double);
56     public:
57         void getData();
58         string toPostfix();
59         double evaluateExpression();
60         int getResult();
61         void output();
62 };
63
64 void Expression::getData()
65 {
66     cout << "Enter an Expression : ";
67     cin >> infix;
68 }
69
70 int Expression::isOperator(char ch)
71 {
72     switch(ch)
73     {
74         case '+': return 1;
75         case '-': return 1;
76         case '*': return 1;
77         case '/': return 1;
78         case '^': return 1;
79         default : return 0;
80     }
81 }
82
83 int Expression::isRightAssociative(char Operator)
84 {
85     return (Operator == '^');
86 }
87
88 int Expression::getPrecedenceValue(char Operator)
89 {
90     switch(Operator)
```

C++ Tab Width: 4 Ln 90, Col 1 INS

Open expression.cpp Save ~ /C++Lab/asn10

```
88 int Expression::getPrecedenceValue(char Operator)
89 {
90     switch(Operator)
91     {
92         case '+' : return 1;
93         case '-' : return 1;
94         case '*' : return 2;
95         case '/' : return 2;
96         case '^' : return 3;
97         default : return 0;
98     }
99 }
100
101 int Expression::hasGreaterPrecedence(char Operator1, char Operator2)
102 {
103     int precedenceValue1 = getPrecedenceValue(Operator1);
104     int precedenceValue2 = getPrecedenceValue(Operator2);
105     if(precedenceValue1 == precedenceValue2)
106     {
107         if(isRightAssociative(Operator1))
108             return 0;
109         return 1;
110     }
111     return (precedenceValue1 > precedenceValue2);
112 }
113
114 int Expression::isOpeningBracket(char ch) // can make these functions inline, by using if and //
condition and operator
115 {
116     switch(ch)
117     {
118         case '(' : return 1;
119         case '{' : return 1;
120         case '[' : return 1;
121         default : return 0;
122     }
123 }
124
125 int Expression::isClosingBracket(char ch)
126 {
127     switch(ch)
128     {
129         case ')' : return 1;
130         case '}' : return 1;
131         case ']' : return 1;
132         default : return 0;
133 }
```

Open expression.cpp Save ~ /C++Lab/asn10

```
125 int Expression::isClosingBracket(char ch)
126 {
127     switch(ch)
128     {
129         case ')': return 1;
130         case '}': return 1;
131         case ']': return 1;
132         default : return 0;
133     }
134 }
135
136 string Expression::toPostfix()
137 {
138     Stack<char> s;
139     string postfix;
140     for(size_t i = 0; i < infix.size(); ++i)
141     {
142         if(isdigit(infix[i]))
143             postfix.push_back(infix[i]);
144         else if(isOperator(infix[i]))
145         {
146             while(!s.isEmpty() && !isOpeningBracket(infix[i]) && hasGreaterPrecedence(s.Top(),
infix[i]))
147                 postfix.push_back(s.pop());
148             s.push(infix[i]);
149         }
150         else if(isOpeningBracket(infix[i]))
151             s.push(infix[i]);
152         else if(isClosingBracket(infix[i]))
153         {
154             while(!s.isEmpty() && !isOpeningBracket(s.Top()))
155                 postfix.push_back(s.pop());
156             s.pop();
157         }
158     }
159     while(!s.isEmpty())
160         postfix.push_back(s.pop());
161     return postfix;
162 }
163
164 double Expression::performOperation(char Operator, double operand1, double operand2)
165 {
166     switch(Operator)
167     {
168         case '+': return (operand1 + operand2);
169         case '-': return (operand1 - operand2);
170         case '*': return (operand1 * operand2);
```

C++ Tab Width: 4 Ln 170, Col 1 INS

```
expression.cpp
~/C++Lab/asn10
Save

164 double Expression::performOperation(char Operator, double operand1, double operand2)
165 {
166     switch(Operator)
167     {
168         case '+': return (operand1 + operand2);
169         case '-': return (operand1 - operand2);
170         case '*': return (operand1 * operand2);
171         case '/': return (operand1 / operand2);
172         case '^': return (pow(operand1, operand2));
173         default : return -1;
174     }
175 }
176
177 double Expression::evaluateExpression()
178 {
179     string postfix = toPostfix();
180     Stack<double> s;
181     for(size_t i = 0; i < postfix.size(); ++i)
182     {
183         if(isdigit(postfix[i]))
184             s.push(postfix[i] - '0'); //here the input is integers as of now, but then 5/8 might
            happens which is less than 0, so everything else is double and not int, or else the point value
            is not considered
185         else
186         {
187             double operand2 = s.pop();
188             double operand1 = s.pop();
189             double operand3 = performOperation(postfix[i], operand1, operand2);
190             s.push(operand3);
191         }
192     }
193     result = s.pop();
194     return result;
195 }
196
197 void Expression::output()
198 {
199     cout << "The value of the Expression " << infix << " is : " << result << endl;
200 }
201
202 double getDivisorValue()
203 {
204     double d;
205     cout << "Enter Divisor Value : " << endl;
206     try{
207         cin >> d;
208         if(d == 0)
```



```
expression.cpp
~/C++Lab/asn10

184     s.push(postfix[i] - '0'); //here the input is integers as of now, but then 5/8 might
    happens which is less than 0, so everything else in double and not int, or else the point value
    is not considered
185     else
186     {
187         double operand2 = s.pop();
188         double operand1 = s.pop();
189         double operand3 = performOperation(postfix[i], operand1, operand2);
190         s.push(operand3);
191     }
192 }
193 result = s.pop();
194 return result;
195 }
196
197 void Expression::output()
198 {
199     cout << "The value of the Expression " << infix << " is : " << result << endl;
200 }
201
202 double getDivisorValue()
203 {
204     double d;
205     cout << "Enter Divisor Value : " << endl;
206     try{
207         cin >> d;
208         if(d == 0)
209             throw d;
210     }
211     catch(double)
212     {
213         cout << "Division by zero " << endl;
214         d = getDivisorValue();
215     }
216     return d;
217 }
218
219 int main()
220 {
221     Expression exp;
222     exp.getData();
223     double result = exp.evaluateExpression();
224     exp.output();
225     double d = getDivisorValue();
226     cout << result << " / " << d << " = " << (result / d) << endl;
227     return 0;
228 }
```

```
deven@deven-VirtualBox: ~/C++Lab/asn10
deven@deven-VirtualBox:~/C++Lab/asn10$ g++ expression.cpp
deven@deven-VirtualBox:~/C++Lab/asn10$ ./a.out
Enter an Expression : (9+6-1*4/2^2)
The value of the Expression (9+6-1*4/2^2) is : 14
Enter Divisor Value :
0
Division by zero
Enter Divisor Value :
0
Division by zero
Enter Divisor Value :
4
14 / 4 = 3.5
deven@deven-VirtualBox:~/C++Lab/asn10$
```