

LESSON 5

1 Examples of order evaluation:

Example

1. Write the following equation as a C++ expression:

$$f = \frac{a + b + c + d + e}{10}$$

Solution:

`f = (a + b + c + d + e) / 10;`

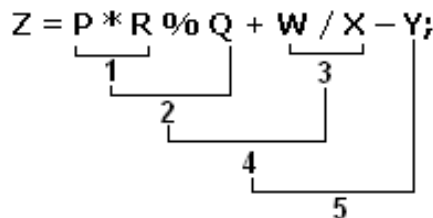
Note: the parentheses here are required because division has a higher precedence than addition.

2. State the order of evaluation for the following expression:

`Z = P * R % Q + W / X - Y;`

Solution:

1. *
2. %
3. /
4. +
5. -



Example

Write C++ program to perform the above equation:

```
#include<iostream.h>
void main( )
{
    int Z, P, R, Q, W, X, Y;
    cout << "enter P:"; cin >> P;
    cout << "enter R:"; cin >> R;
    cout << "enter Q:"; cin >> Q;
    cout << "enter W:"; cin >> W;
    cout << "enter X:"; cin >> X;
    cout << "enter Y:"; cin >> Y;
    Z= P * R % Q + W / X - Y;
    cout << "the result="<< Z;
}
```

LESSON 5

2 The “math.h” Library:

The “math.h” library contains the common mathematical function used in the scientific equations.

Common function from math.h library:	
Mathematical Expression	C++ Expression
e^n	Exp(x)
Log(x)	Log10(x)
Ln(x)	Log(x)
Sin(x)	Sin(x)
x^n	Pow(x,n)
\sqrt{x}	Sqrt(x)

Example:

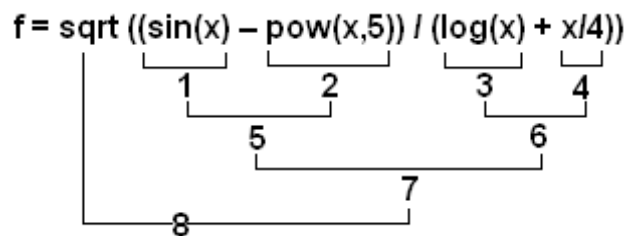
Write the following equation as a C++ expression and state the order of evaluation of the binary operators:

$$f = \sqrt{\frac{\sin(x) - x^5}{\ln(x) + \frac{x}{4}}}$$

Solution:

`f = sqrt ((sin(x) – pow(x,5)) / (log(x) + x/4))`

Order of evaluation:



LESSON 5

The ++ and - - operators can be written either before the variable (prefix notation) or after the variable (postfix notation) as in the following:

Prefix notation:	++ X	X is incremented before its value is taken or returned to current statement.
Postfix notation:	X ++	X is incremented after its value is taken or returned to current statement.

The difference between the Prefix and Postfix notations:

Prefix notation

```
int y;  
int x = 7;  
cout<< ++x <<endl;  
y=x;  
cout<<y;
```

Output:

8
8

Postfix notation

```
int y;  
int x = 7;  
cout<< x++ <<endl;  
y=x;  
cout<<y;
```

Output:

7
8

LESSON 5

3. Manipulator Functions:

They are special stream functions that change certain characteristics of the input and output.

(a) endl: Generate a carriage return or line feed character.

```
cout << "a" << endl;
```

(b) Setbase: It is used to convert the base of one numeric value into another base

Dec(base 10), hex(base 16), oct(base 8)

Example:

1. Write C++ program to convert a base of a number:

```
#include<iostream.h>
void main( )
{
    int value;
    cout << "Enter number:";
    cin >> value;
    cout << "Decimal base="<<dec<<value<<endl;
    cout << "Hexadecimal base="<<hex<<value<<endl;
    cout << "Octal base="<<oct<<value<<endl;
}
```

Output

Enter number: 10 Decimal base=10 Hexadecimal base=a Octal base=12
--

When using **setbase** the program will be:

```
#include <iostream>
#include <iomanip>
void main()
{
    int value;
    cout << "Enter number:";
    cin >> value;
    cout << "Decimal base="<<setbase (10)<<value<<endl;
    cout << "Hexadecimal base="<<setbase (16)<<value<<endl;
    cout << "Octal base="<<oct<<setbase (8)<< value<<endl;
}
```

LESSON 5

(c)Setw: It is used to specify the minimum number of character positions on the O/P field a variable will consume:

Example:

- Write a C++ program to use tab:

```
#include<iostream.h>
#include<iomanip.h>
void main( void)
{
    int a,b;
    a=200;
    b=300;
    cout<<a<<'\\t'<<b<<endl;
}
```

Output

200	300
-----	-----

Example:

- Write C++ program to use setw:

```
#include<iostream.h>
#include<iomanip.h>
void main( void)
{
    int a,b;
    a=200;
    b=300;
    cout<<setw(5)<<a<<setw(5)<<b<<endl;
    cout<<setw(6)<<a<<setw(6)<<b<<endl;
}
```

Output

200 300
200 300

LESSON 5

(d) **Setfill:** It is used to specify a different character to fill the unused field width of the value. **Setfill(char f)**

Example:

- Write C++ program to use setfill:

```
#include<iostream.h>
#include<iomanip.h>
void main( void)
{
    int a,b;
    a=200;
    b=300;
    cout<< setfill('*')<<setw(5)<<a<<setw(5)<<b<<endl;
    cout<<setw(6)<<a<<setw(6)<<b<<endl;
}
```

Output

200300
200300

(e) **Setprecision:** It is used to control the number of digits of an output stream display of a floating point value.

Setprecision (int p)

Example:

- Write C++ program to use setprecision:

```
#include<iostream.h>
#include<iomanip.h>
void main( void)
{
    float a,b;
    a=5.346; b=3.241313;
    cout<<setprecision(3)<<a<< endl;
    cout<<setprecision(6)<<b<< endl;
}
```

Output

5.35
3.24131