

OOP

5/10/21

Lecture : 1

Getting started with C++.

C++ is just an advanced version of C-language with more libraries, functions and features. C-language is structured programming language while C++ is an Object-Oriented Programming language. C++ contains all the features of C (including structural programming).

What is OOP:

OOP stands for Object Oriented programming. Structural programming is all about writing procedures and functions. On the other hand, Object Oriented Programming is about creating objects and classes which contains both i.e data and functions.

Class and Object :

A very basic example of class and object is:

Class : Fruits

Objects : Apple , Mango , Banana.

Further details will be seen in coming lectures.

Sample programs of C++.

1. Hello World

```
#include <iostream> // input/output file
using namespace std; // standard
int main()
{
    // cout is used for printing
    cout << "Hello. World ";
    return 0;
}
```

2. Sum of Two Numbers :

```
#include <iostream>
using namespace std;
int main()
{
    int x,y;
    cout << "Enter 1st no. : ";
    cin >> x;
    cout << "Enter 2nd no. : ";
    cin >> y;
```

```
cout << "Sum is :" << x+y << endl;
return 0;
```

Few of the sample programs
are given below. Get familiar
with the C++ environment
before moving towards the
next lecture.

```
1 #include <iostream>
2
3 using namespace std;
4
5 //Input table no and generate relevant table
6 int main(){
7     int t, i;
8     cout << "Enter table no:";
9     cin >> t;
10
11    for (i=1;i<=10;i++)
12        cout << t << "\t\t" << i << "\t=\t" << t*i << "\n";
13
14    return 0;
15 }
16
```

C:\Users\Umair Javed\Desktop\temp.exe

Enter table no:5

5	x	1	=	5
5	x	2	=	10
5	x	3	=	15
5	x	4	=	20
5	x	5	=	25
5	x	6	=	30
5	x	7	=	35
5	x	8	=	40
5	x	9	=	45
5	x	10	=	50

Process exited after 2.614 seconds with return value 0
Press any key to continue . . .

```
1 #include <iostream>
2
3 using namespace std;
4
5 //Input width and height and draw a box with stars
6 int main(){
7     int width, height;
8
9     cout << "Enter Width:" ;
10    cin >> width;
11    cout << "Enter Height:" ;
12    cin >> height;
13
14    int i, j;
15    //print first line of box
16    for (i=0;i<width;i++)
17        cout << "*";
18    cout << '\n';//to move to next line
19
20    //to print
21    for (i=0;i<height-2;i++)
22    {
23        cout << "* ";
24        //to print spaces between stars
25        for (j=0;j<width-2;j++)
26            cout << ' ';
27
28        cout << "*\n";
29    }
30    //print last line of box
31    for (i=0;i<width;i++)
32        cout << "*";
33
34    cout << '\n';//to move to next line
35
36    return 0;
37 }
```

C:\Users\Umair Java

```
Enter Width:6
Enter Height:5
*****
*   *
*   *
*   *
*****
Process exited a
Press any key to
```

```
1 #include <iostream>
2
3 using namespace std;
4
5
6 int main(){
7     int n, i, j;
8     cout << "Enter N:";
9     cin >> n;
10    for (i=1;i<=n;i++) {
11        //print spaces in start of each line
12        for (j=1;j<=n-i;j++)
13            cout << ' ';
14        //print stars in each line
15        for (j=1;j<=2*i-1;j++)
16            cout << '*';
17        //move to next line
18        cout << '\n';
19    }
20    return 0;
21 }
```

C:\Users\Umair Javed\Desktop\temp.exe

```
Enter N:5
 *
 **
 ***
 ****
 *****
 ******
 ******
 *****
 ****
 ***
 **
 *
```

Process exited after 1.389 seconds with re
Press any key to continue . . .

Self Assessment Lecture - 1

Generate the following outputs .

5

6

7

8

9

8

7

6

5

4

5

6

7

6

5

4

3 4 5 4 3

2 3 2

卷之三

5

10

10

6

1

1

6

4

1

3

3

1

10

7

1

Enter the number of rows: 9

a a a a a a a a a

B B B B B B B B B

c c c c c c c c c

D D D D D D D D D

e e e e e e e e e

F F F F F F F F F

g g g g g g g g g

H H H H H H H H H

i i i i i i i i i

Extra information : (Lec- 1)

- C++ is a super set of C.
- All the standard objects are defined in std namespace.
- << is called stream extraction operator
- >> is called stream insertion operator.
- We can use multiple operators with single cin and cout.

Example :

```
cout << "Hello" << "World" << x;
```

```
cin >> x >> y >> z;
```

- endl stands for endl. It can be used with cout. It

function same as '\n': ie it moves the output to the next line.

Lecture - 2 7/10/21

Dynamic Memory Allocation (DMA) :

Dynamic memory allocation is the memory allocated at run time of variable size. Dynamically allocated memories are accessed by Dynamic arrays or more precisely pointers.

Dynamically allocated memory does not vanishes itself, we need to delete / free the memory manually. The most important use of dynamic memory

allocation is the flexibility as we are free to allocate and deallocate memory whenever we need and even specify the size at runtime.

In C++, we use new and delete operators to access dynamic memory.

New operator :

The new operator denotes a request for memory allocation. If sufficient memory is available, new operator initializes the memory and returns the starting address of the initialized/reserved memory.

Syntax :

pointer-name = new datatype;

OR.

pointer-name = new datatype [size];

Example:

int *P;

(i) P = new int; // OR.

(ii) P = new float[10];

delete operator:

There is a delete operator in C++ to deallocate / delete / free the dynamically allocated memory.

It is programmer's responsibility to manually delete such memory.

Syntax:

delete pointer-name;

OR. delete [] pointer-name;

Example: Deallocating

pointer created in ex: above example.

(i) delete P;

(ii) delete [] P;

What does delete do and what it does not do?

Delete does not delete anything.

Delete does not change the values of ptr (address will still be there)

Delete does not delete the pointer, that is, pointer will still be in scope

Delete will not ensure the deletion of data you stored in Dyn memory. That is upto OS.

Delete only tell the system that I don't need this memory anymore.

Sample program:

```
#include <iostream>
using namespace std;
int main()
{
    int *p, size, i;
    cout << "Enter size : ";
    cin >> size;
    p = new int[size];
    for(i=0; i<size; i++)
    {
        p[i] = i + 10;
        cout << p[i] << ' ';
    }
    delete [] p;
    return 0;
}
```

Few of the sample programs
are given. Get familiar with
DMA before moving further.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main ()
6 {
7     int i,n;
8     int * p;
9     cout << "How many numbers would you like to type? ";
10    cin >> i;
11    p = new int[i];
12
13    for (n=0; n<i; n++)
14    {
15        cout << "Enter number: ";
16        cin >> p[n];
17    }
18    cout << "You have entered: ";
19    for (n=0; n<i; n++)
20        cout << p[n] << ", ";
21    delete[] p;
22
23    return 0;
24 }
```

C:\Users\Umair Javed\Desktop\temp.exe

```
How many numbers would you like to type? 4
Enter number: 23
Enter number: 24
Enter number: 65
Enter number: 75
You have entered: 23, 24, 65, 75,
-----
Process exited after 6.64 seconds with return value 0
Press any key to continue . . .
```

```
1 #include <iostream>
2 #include <cstdlib>           //stdlib.h
3 #include <ctime>
4
5 using namespace std;
6
7 int main(){
8     srand(time(0));
9     #define SIZE 20
10    int i, *p, n, x[SIZE], count = 0, j=0;
11    cout << "Original Array: ";
12    for (i=0;i<SIZE;i++){
13        x[i] = rand() % 100;
14        cout << x[i] << ' ';
15    }
16    cout << "\n\nNew Array Less than 50: ";
17    for (i=0;i<SIZE;i++)
18        if (x[i] < 50) count++;
19    p = new int[count];
20    for (i=0;i<SIZE;i++)
21        if (x[i] < 50) p[j++] = x[i];
22
23    for (i=0;i<count;i++)
24        cout << p[i] << ' ';
25
26    delete[] p;
27
28    return 0;
29}

```

C:\Users\Umair Javed\Desktop\temp.exe

Original Array: 51 45 52 54 5 4 85 0 95 81 73 43 71 47 85 66 52 87 66 94

New Array Less than 50: 45 5 4 0 43 47

Process exited after 0.0361 seconds with return value 0

Press any key to continue . . .

Self Assessment :

Lecture - 2 : DMA

- (i) Take size and then values from user and find the largest number int int.
- (ii) ~~Take~~ Take size from user. Generate random numbers in range 20 - 40 (both inclusive). Sort them using Selection sort.
- (iii) Take size from user. generate random numbers from 2-99 (both inclusive) • Count the number of prime numbers in it • Now create another array of (size - count) and store all numbers except primes in it.

Wise Words

Extra Information :

Lec-2 : DMA

- Dynamic memory is created in Heap while static memory is created in stack. (in C++)
- There are two methods to check if memory is successfully allocated because there is always a possibility that user can give much larger value than the free space.
- new operator can also be used to initialize the variable by using parenthesis. i.e.
`int *p ;`

`P = new int(5) ;`

Now, P is created and also initialized by 5. Remember, don't confuse it with brackets

that are `[]`. Here, we used `()`. Brackets `[]` are used to allocate a block of memory (an array).

- `new` operator can allocate memory of any data. It means, we can allocate memory of not only built-in data-types (`int`, `float`, `double`), but we can also allocate an array of user defined datatypes (defined by using structs or classes).
- There is a difference between normal array and dynamically allocated array. Normal arrays are automatically deallocated by compiler (when the function ends). However, dynamically allocated arrays always remain there until they are

- deallocated by programmer.
- If programmer does not deallocate a memory it causes memory leak.

File Handling in C++

File handling is a technique of storing output and getting input from a file. We use file handling to store the output of a program permanently. C++ supports file handling and provides us three main classes for the purpose.

i.e.

`ofstream`

`ifstream`

`fstream`.

Note: We need to add a header file in our program.

#include <fstream>

ofstream:

It is used to write data to a file.

If the file does not exists, it ~~creates the file~~ can be used to create a new file.

ifstream:

It is used to read data from a file. It is also known as input stream. It does not create a new file.

fstream:

It is a combination of both. It provides the capability of reading, writing and creating the files.

Remember : We need to close the file before ending the program.

Syntax:

class-stream pointer-name("name.ext")

Example:

```
ofstream file("numbers.txt");
ifstream fp ("\.\numbers.txt");
fstream num ("D:\number.txt");
```

Sample Programs:

Lec-2 : Files

Few sample programs of File handling are given.
Get familiar with input/output in files before moving further.

```
1 #include <iostream>
2 #include <fstream>
3 #include <cstdlib>
4 #include <ctime>
5
6
7
8 //File handling in C++
9
10 int main(){
11     //Read numbers from file numbers.txt
12     int i, n;
13     ifstream input ("numbers.txt");
14     input >> n;
15     cout << n << '\n';
16     input >> n;
17     cout << n << '\n';
18     input >> n;
19     cout << n << '\n';
20     input >> n;
21     cout << n << '\n';
22
23     input.close();
24     return 0;
25 }
```

```
1 #include <iostream>
2 #include <fstream>
3
4 using namespace std;
5
6 int main(){
7     int i, n, sum=0, *x;
8     ifstream input("numbers.txt");
9     input >> n;
10    x = new int[n];
11    for (i=0;i<n;i++){
12        input >> x[i];
13        sum = sum + x[i];
14    }
15    float avg = (float)sum / n;
16    cout << "Less than Avg: ";
17    for (i=0;i<n;i++)
18        if (x[i] < avg)
19            cout << x[i] << ' ';
20    cout << '\n';
21    cout << "Greater than Avg: ";
22    for (i=0;i<n;i++)
23        if (x[i] > avg)
24            cout << x[i] << ' ';
25    cout << '\n';
26    return 0;
27 }
```

numbers.txt - Notepad

File Edit Format View Help

|10

56 45 53 65 12 36 45 85 32 14

```
C:\Users\Umair Javed\Desktop\temp.exe
Less than Avg: 12 36 32 14
Greater than Avg: 56 45 53 65 45 85
-----
Process exited after 0.04502 seconds w
Press any key to continue . . .
```

SELF ASSESSMENT :

Lec - 2 : File Handling

- (i) Write a program to count the number of lines in a file.
- (ii) Write a program to duplicate any file.
- (iii) Open any .c or .cpp file in your program and make a copy of the code in the new file but without comment.
(i.e. You need to remove comments from a code file)
Hint : ignore every line starting with // .

Extra Information :

Lec - 2 : File Handling

- There are four functions we can apply on every file.

`open();`

`close();`

~~`read();`~~

`write();`

- Alternative Syntax :

`ofstream/fifstream/Fstream bointer-name;`

`Pointername.open("filename", mode);`

Example :

```
fstream fp;
```

```
fp.open("number.txt", ios::out);
```

- There are total of eight of modes in which file can be opened in C++.

(Google for details).

- All file opening modes are defined in `iso` namespace.
- `iso::in`, `iso::out`, `iso::app` are most common file opening modes for read, write and append respectively.