

Object Oriented Programming using C++(Unit-1)

- Program structure
- Namespace
- Identifiers
- Variables
- Constants
- Enum
- Operators



Program structure

- Include header/source files
- Class declaration
- Class member functions definition
- Main function



Namespace

- Consider the simple C++ program-

```
#include <iostream>
using namespace std;
int main ()
{
    cout << "Hello World!";
    return 0;
}
```

Output:-Hello World!



Namespace

- It defines a scope of the identifiers(variables, functions, classes etc.) that are used in a program.
- using directive includes the namespace in the program.
- std is the namespace where ANSI C++ standard class libraries are defined.
- It will bring all the identifiers defined in std to the current global scope.
- using and namespace are keywords of C++.



Identifiers

- Identifiers refers to the name of variable , functions, array, class etc. created by programmer.
- Rules for naming the identifiers-
 1. Only letters, digits and underscore are permitted.
 2. The name can't start with a digit.
 3. Upper case and lower case letters are distinct.
 4. A keyword can't be used as a identifier.
- In ANSI C the maximum length of a variable is 32 chars but in C++ there is no bar.



Variables

- Variables in C++ is a name given to a memory location. It is the basic unit of storage in a program.
- The value stored in a variable can be changed during program execution.
- Rules For Declaring Variable
 1. The name of the variable contains letters, digits, and underscores.
 2. The name of the variable is case sensitive.
 3. The name of the variable does not contain any whitespace and special characters.
 4. All the variable names must begin with a letter of the alphabet or an underscore(_).
 5. We cannot use C++ keyword as a variable name.

Variables

- Variable declaration-

- Example-

```
int time, speed; // Declaring integer variables  
speed = 80; //Variable initialization
```

- Variable definition-

- Example-

```
double price = 50.5; //Variable declaration with  
initialization
```




Constants

- Constants are fixed values that do not change during the program execution.
- There are two categories of constants-
 1. Literal constants
 2. Symbolic constants
- Literal constants are of four types-
 1. Integer constants like 56, -35(decimal), 064(octal), 0x9F(hexadecimal)
 2. Real constants like -0.75, 0.64e3, 13.5

Constants

3. Character constants like 'A', '8', '\$'
4. String constants like "Ajmer", "2022"
- There are two ways of creating symbolic constants in C++.
1. Using the qualifier const.

```
const int size = 10 ;
```
- It creates typed constants instead of having to use #define to create constants that have no type information.
- C++ requires a const to be initialized.

Constants

2. using enum keyword .

- Example-

```
enum {X, Y , Z};
```

- It defines X, Y and Z as integer constants with values 0,1 and 2 respectively.
- The values can be assigned explicitly.
- Example-

```
enum {X=5, Y=10, Z=15};
```


Enum

- An enumerated data type is an user defined type which provides a way for attaching names to number.
- **enum** keyword automatically enumerates a list of words by assigning them values 0,1,2 and so on.
- It provides an alternative means for creating symbolic constants.
- Example-

```
enum shape { circle, square, triangle} ;  
enum colour{red=4, blue, green, yellow};  
enum position {off, on};
```

Enum

- In C++, the tag names shape, colour, and position become new type names.
- Example-

```
colour background ; // background is of type colour
background = red;
cout<<background; //print 4
cout<<yellow; //print 7
```


Enum

- C++ does not allow an int value to be automatically converted to an enum.
- Example-
 `colour background =7; //error in c++`
 `colour background =(colour) 7;//valid`
- An enumerated value can be used in place of an int value.
- Example-
 `int c=blue ;//valid, colour type promoted to int`
 `cout<<c;// print 5`

Operators

- All C operators are valid in C++ also. In addition C++ introduces some new operators.

➤ <<	insertion operator
➤ >>	extraction operator
➤ ::	scope resolution operator
➤ ::*	pointer to member declarator
➤ ->*	pointer to member operator
➤ .*	pointer to member operator
➤ new	memory allocation operator
➤ delete	memory release operator

Operators

- Insertion operator(<<) - It directs the information on its right to the object on its left.
- Example-

```
cout<<"Hello"; // print Hello
```

```
cout<<variable; // print value of variable
```

- Extraction operator(>>) - It extracts value from the keyboard and assigns it to the variable on its right.
- Example-

```
char str[10];
```

```
cin>>str; // read a string and store in str
```

variable

Operators

- Scope resolution operator(::) – It allows access to the global version of a variable.
- Example-

```
int m=10;
main()
{
    int m=20;
    {
        int m=30;
        cout<<"m="<<m<<endl;
        cout<<":: m="<<:: m<<endl;
    }
    cout<<"m="<<m<<endl;
    cout<<":: m="<<:: m<<endl;
}
```


Operators

- Output-

m=30

::m=10

m=20

::m=10

- Its major application is in the classes to identify the class to which a member function belongs.

Operators

- Member dereferencing operators- These operators used to access the class members through pointers. These three operators are-
 1. Pointer to member declarator(`::*`)- To declare a pointer to a member of a class.
 2. Pointer to member operator(`->*`)- To access a member using a pointer to the object and a pointer to that member.
 3. Pointer to member operator(`.*`)- To access a member using a object name and a pointer to that member.

Operators

- Memory allocation operator(new)- It allocates memory at run time.
- Example-

`int *p=new int; // allocates memory for integer variable`

`*p=25; // Assign 25 to the newly created int object`

- Memory release operator(delete)- It deallocates memory at run time.
- Example-

`delete p; // deallocates memory allocated to pointer p`

- If a data object is no longer needed, it is destroyed to release the memory space for reuse.



Thank You

Prepared by: Anil Kumar Tailor, Asst. Prof. Engineering College
Ajmer