Content 4

Variable Scope & Data Types in C++

* **Variable Scope**
* **Data Types**

Before explaining the concept of variable scope, I would like to clarify about variables a little more. Variable can be defined as a container to hold data. Variables are of different types, for example:

1. **Int**-> Int is used to store integer data e.g (-1, 2, 5,-9, 3, 100).
2. **Float**-> Float is used to store decimal numbers e.g (0.5, 1.05, 3.5, 10.5)
3. **Char**-> Char is used to store a single character e.g. ('a', 'b',' c', 'd')
4. **Boolean**-> Boolean is used to store "true" or "false."
5. **Double**-> Double is also used to store decimal numbers but has more precision than float, e.g. (10.5895758440339...)

Here is an example to understand variables: int sum = 34; means that sum is an integer variable that holds value '34' in memory.

**Syntax for Declaring Variables in C++**

Data\_type Variable\_name = Value;

**Example:**

 int a = 44, b = 10;

 char letter = 'A';

#### Variable Scope

The scope of a variable is the region in the program where the existence of that variable is valid. For example, consider this analogy - if one person travels to another country illegally, we will not consider that country as its scope because he doesn't have the necessary documents to stay in that country.

Base on scope, variables can be classified into two types:

* **Local variables**
* **Global variables**

#### Local variables:

Local variables are declared inside the braces of any function and can be assessed only from there.

**Example:**

#include <iostream>

using namespace std;

int main()

{

    int a=4;       // Local Variable

    cout<<"The value of a is: "<<a;

    return 0;

}

Here a is local variable because it is declared in main functions body.

#### Global variables:

Global variables are declared outside any function and can be accessed from anywhere.

#include <iostream>

using namespace std;

int a=4;       // Global Variable

int main()

{

    cout<<"The value of a is: "<<a;

    return 0;

}

Here a is now Global Variable because it is declared and initialized out of main body.

Here if you declare both Global and Local Variable Who will Take precidency?

Local variable will take the presidency.

Code for Explaining the Global and Local Variable Presidency:

#include <iostream>

using namespace std;

int a=4;       // Global Variable

int main()

{   int a=10;   //Local Variable

    cout<<"The value of a is: "<<a;

    return 0;

}

**Output:**

The value of a is: 10

Here If I have to give presidenc to the global Variable than I have to use this **‘::’**

**Code for giving the presidency to the global variable in the presence of Local Variable:**

int a=4;       // Global Variable

int main()

{   int a=10;   //Local Variable

    cout<<"The value of a is: "<<::a;  //here i have gaved presidence to global variable

    return 0;

}

**Output:**

The value of a is: 4

#### Data Types

Data types define the type of data that a variable can hold; for example, an integer variable can hold integer data, a character can hold character data, etc.

Data types in C++ are categorized into three groups:

* **Built-in**
* **User-defined**
* **Derived**

##### **1. Built-in Data Types in C++:**

* Int
* Float
* Char
* Double
* Boolean

##### **2. User-Defined Data Types in C++:**

* Struct
* Union
* Enum

Note: We will discuss the concepts of user-defined data types in another Content. For now, understanding that these are user-defined data types is enough.

##### **3. Derived Data Types in C++:**

* Array
* Pointer
* Function

Note: We will discuss the concept of derived data types in another Content. For now, understanding that these are derived data types is enough.

**Code for Underrstand Built In data type:**

#include <iostream>

using namespace std;

int main()

{   int a=10;

    char b='G';

    float PI=1.3;

    bool y=true; //Whenever it is true it will give 1

    bool z=false; //Whenever it is false it will give 0

    cout<<"The value of a is: "<<a<<endl;

    cout<<"The value of b is: "<<b<<endl;

    cout<<"The value of PI is: "<<PI<<endl;

    cout<<"The value of y is: "<<y<<endl;

    cout<<"The value of z is: "<<z<<endl;

    return 0;

}

**For Any Boolean value**; if it is true it will return 0 and if false then return 1.

**Output:**

The value of a is: 10

The value of b is: G

The value of PI is: 1.3

The value of y is: 1

The value of z is: 0