

## Data Types and Variables

While writing program in any language, you need to use various variables to store various information. Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

You may like to store information of various data types like character, wide character, integer, floating point, double floating point, boolean etc. Based on the data type of a variable, the operating system allocates memory and decides what can be stored in the reserved memory.

## Primitive Built-in Types

C++ offers the programmer a rich assortment of built-in as well as user defined data types. Following table lists down seven basic C++ data types.

Type	Keyword
Boolean	bool
Character	char
Integer	int
Floating point	float
Double floating point	double
Valueless	void
Wide character	wchar_t

Several of the basic types can be modified using one or more of these type modifiers:

- signed
- unsigned
- short
- long

Implement the following in a new project, within a file titled **DataTypes.cpp**, and then build and execute your program.

```
#include <iostream>
using namespace std;

int main() {
    cout << "Size of char : " << sizeof(char) << endl;
    cout << "Size of int : " << sizeof(int) << endl;
    cout << "Size of short int : " << sizeof(short int) << endl;
    cout << "Size of long int : " << sizeof(long int) << endl;
    cout << "Size of float : " << sizeof(float) << endl;
    cout << "Size of double : " << sizeof(double) << endl;
    cout << "Size of wchar_t : " << sizeof(wchar_t) << endl;

    return 0;
}
```

The **sizeof()** operator to get size of various data types.

## Local Variables

Variables that are declared inside a function or block are local variables. They can be used only by statements that are inside that function or block of code. Local variables are not known to functions outside their own.

Implement the following in a new project, within a file titled **LocalVariables.cpp** and execute your program.

```
#include <iostream>
using namespace std;

int main () {
    // Local variable declaration:
    int a, b;
    int c;

    // actual initialization
    a = 10;
    b = 20;
    c = a + b;

    cout << c;

    return 0;
}
```

## Global Variables

Global variables are defined outside of all the functions. The global variables will hold their value throughout the life-time of your program. A global variable can be accessed by any function. That is, a global variable is available for use throughout your entire program after its declaration.

Implement the following in a new project, within a file titled **GlobalVariables.cpp**, and then build and execute your program

```
#include <iostream>
using namespace std;

// Global variable declaration:
int g;

int main () {
    // Local variable declaration:
    int a, b;

    // actual initialization
    a = 10;
    b = 20;
    g = a + b;

    cout << g;

    return 0;
}
```

A program can have same name for local and global variables but value of local variable inside a function will take preference.

Modify your **GlobalVariables.cpp** file as follows, and then build and execute your program.

```
#include <iostream>
using namespace std;

// Global variable declaration:
int g = 20;

int main () {
    // Local variable declaration:
    int g = 10;

    cout << g;

    return 0;
}
```

## Constants

You can use **const** prefix to declare constants with a specific type.

Implement the following in a new project, within a file **Constants.cpp** and then build and execute your program.

```
#include <iostream>
using namespace std;

int main() {
    const int LENGTH = 10;
    const int WIDTH = 5;
    const char NEWLINE = '\n';
    int area;

    area = LENGTH * WIDTH;
    cout << area;
    cout << NEWLINE;
    return 0;
}
```

## References

- [https://www.tutorialspoint.com/cplusplus/cpp\\_data\\_types.htm](https://www.tutorialspoint.com/cplusplus/cpp_data_types.htm)
- [https://www.tutorialspoint.com/cplusplus/cpp\\_constants\\_literals.htm](https://www.tutorialspoint.com/cplusplus/cpp_constants_literals.htm)