

Data Handling

Basic Data Types

C++ supports a large number of data types. The built in or basic data types supported by C++ are integer, floating point and character. C++ also provides the data type bool for variables that can hold only the values True and false.

Some commonly used data types are summarized in table along with description.

Type	Description
int	Small integer number
long int	Large integer number
float	Small real number
double	Double precision real number
long double	Long double precision real number
char	A Single Character

The exact sizes and ranges of values for the fundamental types are implementation dependent. The header files `<climits>` (for the integral types) and `<cfloat>` (for the floating-point types) specify the ranges of values supported on your system.

C++ string Class

Because a `char` variable can store only one character in its memory location, another data type is needed for a variable able to hold an entire string. While C++ does not have a builtin data type able to do this, Standard C++ provides string class that allows the programmer to create a string type variable.

in order to declare and use objects (variables) of this type we need to include an additional header file in our source code: <string>

```
// This program demonstrates the string class.
#include <iostream>
#include <string> // Required for the string class. using
namespace std;

int main ()
{
    string mystring = "This is a
string";
    cout << mystring;
    return 0;
}
```

Variable Initialization

Variable is a location in the computer memory which can store data and is given a symbolic name for easy reference. The variables can be used to hold different values at different times during the execution of a program.

Declaration of a variable

Before a variable is used in a program, we must declare it. This activity enables the compiler to make available the appropriate type of location in the memory.

```
float total;
```

You can declare more than one variable of same type in a single single statement `int x, y;`

Initialization of variable

When we declare a variable it's default value is undetermined. We can declare a variable with some initial value.

```
int a = 20;
```

The other way to initialize variables, known as constructor initialization, is done by enclosing the initial value between parentheses () :

For example: `int`

```
a (0);
```

Both ways of initializing variables are equivalent in C++.

Constants

A variable which does not change its value during execution of a program is known as a constant variable. Any attempt to change the value of a constant will result in an error message. A constant in C++ can be of any of the basic data types, `const` qualifier can be used to declare constant as shown below:

```
const float PI = 3.1415;
```

The above declaration means that PI is a constant of float types having a value 3.1415.

Examples of valid constant declarations are:

```
const int RATE = 50; const  
float PI = 3.1415;  
const char CH = 'A';
```

Type Conversion

The process in which one pre-defined type of expression is converted into another type is called conversion. There are two types of conversion in C++.

1. Implicit conversion
2. Explicit conversion

Implicit conversion

Data type can be mixed in the expression. For example

```
double a; int  
b = 5; float  
c = 8.5; a =  
b * c;
```

When two operands of different type are encountered in the same expression, the lower type variable is converted to the higher type variable. The following table shows the order of data types.

Order of data types	
Data type	order
long double	(highest)
double	
float	
long	To
int	
char	(lowest)

The int value of b is converted to type float and stored in a temporary variable before being multiplied by the float variable c. The result is then converted to double so that it can be assigned to the double variable a.

Explicit conversion

It is also called type casting. It temporarily changes a variable data type from its declared data type to a new one. It may be noted here that type casting can only be done on the right hand side the assignment statement.

```
totalPay = static cast<double>(salary) + bonus;
```

Initially variable salary is defined as float but for the above calculation it is first converted to double data type and then added to the variable bonus.