

Creating variables

A “variable” is like a container in a C++ program in which a data value can be stored inside the computer’s memory. The stored value can be referenced using the variable’s name.

The programmer can choose any name for a variable providing it adheres to the C++ naming conventions – a chosen name may only contain letters, digits, and the underscore character, but cannot begin with a digit. Also the C++ keywords, listed on the inside cover of this book, must be avoided. It’s good practice to choose meaningful names to make the code more comprehensible.

To create a new variable in a program it must be “declared”, specifying the type of data it may contain and its chosen name. A variable declaration has this syntax:

data-type variable-name ;

Multiple variables of the same data type can be created in a single declaration as a comma-separated list with this syntax:

data-type variable-name1 , variable-name2 , variable-name3 ;

The five basic C++ data types are listed in the table below, together with a brief description and example content:

Data Type:	Description:	Example:
char	A single byte, capable of holding one character	'A'
int	An integer whole number	100
float	A floating-point number, correct to six decimal places	0.123456
double	A floating-point number, correct to ten decimal places	0.0123456789
bool	A boolean value of true or false , or numerically zero is false and any non-zero is true	false or 0 true or 1

Variable declarations must appear before executable statements – so they will be available for reference within statements.

Beware



Names are case-sensitive in C++ – so variables named **VAR**, **Var**, and **var** are treated as three individual variables. Traditionally C++ variable names are lowercase and seldom begin with an underscore as some C++ libraries use that convention.

Beware



Character values of the **char** data type must always be enclosed between single quotes – not double quotes.

...cont'd

When a value is assigned to a variable it is said to have been “initialized”. Optionally a variable may be initialized in its declaration. The value stored in any initialized variable can be displayed on standard output by the `cout` function, which was used on page 12 to display the “Hello World!” greeting.

- 1 Start a new program by specifying the C++ library classes to include and a namespace prefix to use

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

- 2 Add a main function containing a final return statement

```
int main()
{
    // Program code goes here.
    return 0 ;
}
```

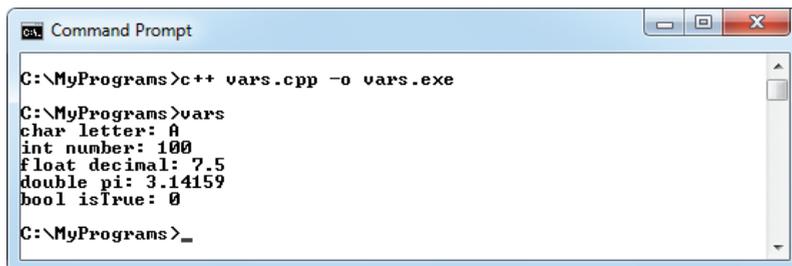
- 3 In the main function, insert statements to declare and initialize variables of various data types

```
char letter ;    letter = 'A' ;    // Declared then initialized.
int number ;    number = 100 ;    // Declared then initialized.
float decimal = 7.5 ;    // Declared and initialized.
double pi = 3.14159 ;    // Declared and initialized.
bool isTrue = false ;    // Declared and initialized.
```

- 4 Now insert statements to output each stored value

```
cout << "char letter: " << letter << endl ;
cout << "int number: " << number << endl ;
cout << "float decimal: " << decimal << endl ;
cout << "double pi: " << pi << endl ;
cout << "bool isTrue: " << isTrue << endl ;
```

- 5 Save, compile, and run the program to see the output



```
Command Prompt
C:\MyPrograms>c++ vars.cpp -o vars.exe
C:\MyPrograms>vars
char letter: A
int number: 100
float decimal: 7.5
double pi: 3.14159
bool isTrue: 0
C:\MyPrograms>_
```



vars.cpp

Hot tip



Always begin boolean variable names with “is” so they are instantly recognizable as booleans. Also, use “lowerCamelCase” for all variable names that comprise multiple words – where all except the first word begin with uppercase, like “isTrue”.