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:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>A single byte, capable of holding one character</td>
<td>‘A’</td>
</tr>
<tr>
<td>int</td>
<td>An integer whole number</td>
<td>100</td>
</tr>
<tr>
<td>float</td>
<td>A floating-point number, correct to six decimal places</td>
<td>0.123456</td>
</tr>
<tr>
<td>double</td>
<td>A floating-point number, correct to 10 decimal places</td>
<td>0.0123456789</td>
</tr>
<tr>
<td>bool</td>
<td>A Boolean value of true or false, or numerically zero is false and any non-zero is true</td>
<td>false or 0, true or 1</td>
</tr>
</tbody>
</table>

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

1. Start a new program by specifying the C++ library classes to include, and a namespace prefix to use
   ```cpp
   #include <iostream>
   using namespace std;
   ```

2. Add a main function containing a final `return` statement
   ```cpp
   int main()
   {
       // Program code goes here.
       return 0;
   }
   ```

3. In the main function, insert statements to declare and initialize variables of various data types
   ```cpp
   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
   ```cpp
   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
   ```cmd
   C:\MyPrograms>g++ 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
   ```

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”.

...cont’d