I. PROGRAM STRUCTURE

```
// A First Program

#include <iostream>
using namespace std;

int main()
{
  cout << "Hello World\n";
  return 0;
}
```

// indicates that the remainder of the line is a comment

#include <iostream>
: is a preprocessor directive. Lines which begin with # are processed before the program is compiled. In this case, the preprocessor is told to include input/output stream information in the std namespace. All programs the use keyboard/screen I/O need this information.

using namespace std;
: Lets the compiler know that the std namespace is being used. Both directives And namespaces will be addressed in detail in Chapter 8.

int main()
: begins the main function, which is a part of every C++ program. A C++ program may consist of many functions, but the main function is called first.

{ ... }
: enclose the function body, that is, the statements which comprise the function

cout << "Hello World\n";
: causes the string "Hello World" to be printed to the screen, followed by a carriage return (\n). The ";" is a statement terminator.

return 0;
: as all functions return a value, the main function does also. This statement specifies that a 0 will be returned by the function.

** It is good programming practice to indent the entire body of a function, to help indicate where the function begins and ends.

II. VARIABLE DECLARATIONS

In C++, a variable is used to store data values. A variable is a named memory location.

A variable DECLARATION reserves memory for storage. A variable declaration looks like this:

```
DATA_TYPE   VARIABLE_IDENTIFIER;
```

where DATA_TYPE is a C++ data type (int, float, double, char, etc).and VARIABLE_IDENTIFIER is the name being given to the memory being reserved.

For example,

```
int answer;
```

reserves one memory location for a value of type integer, and this location is called ANSWER. More than one location of the same type can be declared at the same time as:

```
int  result, average, score;
```

which reserves three memory locations for storing decimal values.

Guidelines for Picking Names:

* may contain letters, digits and underscores
* may NOT begin with a digit
* C++ is case sensitive, Abc is different than ABC.
* use fewer than 32 characters for portability between compilers

III. USING A VARIABLE

An assignment statement is used to store a value in a variable that has been declared:

```
answer = 5;                  // stores the value 5 in the location 'answer'

result = answer + 5;         // adds 5 to value in 'answer' and stores sum in 'result'

score = result + average  // will add the value of 'result' to the value of 'average',
                             // storing the sum in 'score'
```

** if average has not been explicitly assigned a value, whatever happens to be in that memory location is used for the calculati

** if a decimal value is assigned to an integer variable, the decimal value is truncated

** if an integer value is assigned to a float variable, a floating point representation is stored

Arithmetic operators include:

+ addition
- subtraction
* multiplication
/ division (if two both operands are integer, integer division is performed)
% modulus (the remainder of integer division)

If more than one operator is used in an expression, evaluation is done from left to right, with respect to operator precedence:

( ) expressions enclosed in parentheses have highest precedence
*, / or % multiplicative operations are performed after parethesized stuff
+, - addition and subtraction have lowest precedence

Shorthand notation provides an easy way to perform an operation on a variable.

answer += 5; is equivalent to answer = answer + 5;

IV. STREAM I/O

As seen above, the object 'cout' is used for stream output (to the screen). Cout belongs to the namespace 'std'. If we do not specify the namespace with the USING directive

as (as in the sample above), we must reference cout as " std::cout. cout is used with the stream insertion operator, ' << '. When this operator is used, the value to the right of the operator is inserted into the output stream (cout).

The input stream object is 'cin'. It used with the stream extraction operator ' >> '. When this operator is used, character input is taken from the input stream and stored in the variable named on the right of the operator, as seen in the example below, which reads and echoes an input value.

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

 int main()
{
      int number;

      cout << "Please enter an integer value\n";
      cin >> number;
      cout << "The value you entered is: " << number ;
      return 0;
   }
```

V. C++ DATA TYPES

Fundamental data types in C++ include:

Integer data types (int, short, long)
Decimal data types (float,double, long double)
character data types (char)
boolean data type (bool)