Sample C++ Program

// A simple C++ program
#include <iostream>
using namespace std;

int main ()
{
    cout << "Programming is fun";
    return 0;
}
Parts of a C++ Program

- **Comment**: //...
  - ignored by compiler
  - notes to human reader
- **Preprocessor Directive**: #include ...
  - preprocessor inserts contents of file here
- **using namespace std**;
  - every name in iostream is part of std namespace

- **int main ()**
  - start of function (group of stmts) named main
  - the starting point of the program
- **{ }**
  - contains the body of the function
- **cout << “Programming is fun”;**
  - statement that displays message on screen
- **return 0;**
  - sends value of 0 to OS (means success!)
Literals

- A literal represents a constant value from a given data type.
- Numbers: 0, 34, 3.14159, -1.8e12, etc.
- Characters: 'A', 'z', '!', '5', etc.
- Strings (sequence of characters):
  - “Hello”, “This is a string”
  - “100 years”, “100”, “Y”, etc.
- NOTE: These are all different: 5, '5', “5”

Special characters

- Newline: '\'n'
- Double quote: '\'"
- These can occur in strings:
  - “hello\nthere”
  - “she said "boo" very quietly”
- See textbook for more
- It's a backslash (\), not a slash (/)
The cout object

- **cout**: console output (represents the screen)
  - a stream object: works on a sequence of data
- **<<**: the stream insertion operator
  - sends value on right-hand side (rhs) to stream on left-hand side (lhs)
  - `cout << “This is an example.”`
- **endl**: a stream manipulator
  - advances output to start of next line

cout: examples

```cpp
cout << “This is an example.”;
cout << “This is” << “ an example.”;
cout << “This is”;
cout << “ an example”;
```

```cpp
cout << “The best selling book on Amazon”;
cout << “ is "The Help"”;
cout << “The best selling book on Amazon” << endl;
cout << “ is "The Help"”;
cout << “The best selling book on Amazon\n is "The Help"”;
```
Identifiers

- An identifier is a name for some program element.
- Rules:
  - May not be a keyword (see p. 43)
  - First character must be a letter or underscore
  - Following characters must be letters, numbers or underscores.
- Identifiers are case-sensitive:
  - myVariable is not the same as MyVariable

Variables

- Variable: named location in storage (memory)
- Variable definition in a program:
  - <datatype> <identifier>;
- examples:
  - int someNumber;
  - char firstLetter;
Assignment statement

- Stores or saves a value to a variable
- \(<\text{variable}> = <\text{rhs}>\)
- Value of the rhs is stored in the variable
- examples:
  - `someNumber = 100;`
  - `firstLetter = 'J';`

Program with a variable

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

int main()
{
    int number;
    number = 100;
    cout << "The value of the number is " << number << endl;
    return 0;
}
```