C++ Basic Elements

To begin programming we will first look at the basic components that make up a simple program.

This foundation will become the structure for each of the programs in your course and will help you grow to use more complex elements of the C++ language.

Comments

While they execute NOTHING when it comes to the code, comments are probably some of the most important lines you will add to your program.

Comments allow you to remember how you structured your program from day to day.

It will also allow others to know how your program works for debugging and collaborative purposes.

There are two ways to comment a program:

```
// program.cpp
// This program does a lot of stuff!!
```

The use of the double slash will indicate that a line is a **comment** and will be ignored by the compiler. This will only work for that particular line!

If you wish to comment multiple lines you can start each one with the double slash or:

```
/* This will begin a long series of comments that will help explain the
function and variables in the program. It will be terminated by using
the reverse of the symbol. */
```

Compiler Directive #include

The first line of most programs that is not a comment is:

```
#include <iostream>
```

This line loads predefined commands from the iostream class before the program even begins to compile or run.

A **class** or a **library** is a standard set of commands that has already been written and tested. This allows us to reuse common commands.

The class **iostream** enables our program to read input data from the keyboard and display the results on the screen.

There are many files such as this we will make use of throughout the course.

Namespace std

The line:

```
using namespace std;
```

indicates that we will be using objects that are named in a special reason *namespace std* (standard). We will always be working in this space and it will appear in all of our problems.

Function main

The line

```
int main()     // start of function main
```

indicates the start of the function main, where the program begins.

A **function** is a collected of related statements and commands under the same name.

Every program must have a main function. The word **int** indicates that the program should return an integer value when complete.

Each function must be enclosed in curly braces { }.

Declaration Statements

A function body consists of two kinds of statements **declaration statements** and **executable statements**.

Declaration statements tell the compiler what data is needed in the function.

```
const float KM_PER_MILE = 1.609;          // 1.609 km in a mile
float miles, kms;
```

Executable statements cause some action to take place when the program is run.

```
cout << "Enter the distance in miles: ";
cin >> miles;
```

Reserved Words and Identifiers

An identifier is a label we will place on a value or a command in order to manipulated by the program.

There are special reserved words that CAN'T be used. Just some examples of these **reserved words** or **keywords** are:

const, float, include, int, namespace, return, using

Otherwise identifies need to follow these rules:

1) An identifier must always begin with a letter or underscore symbol (not recommended)

2) An identifier must consist of letters, digits, or underscores only.

3) You can't use reserved words as an identifier.

Some valid identifiers are:

letter1, Letter1, letter2, inches, cent, centPerInch, cent_per_inch, hello

Some invalid identifiers are:

1Letter, float, const, Two*Four, Joe's, two-dimensional