C++ Structs

Recall that arrays can be used to store large amounts of related information. However, the information stored in arrays must be homogeneous, i.e. must be all of the same type. Thus, arrays cannot handle all program storage needs. Suppose for example, that we want to write a program for a business that keeps customer information. Then we might need to store information such as

```
Name            (string or character array)
CustomerNumber  (long)
BalanceDue      (float)
```

Structs allow us to create variables that can contain related variables of different data types. A struct to hold the information above might be defined as follows:

```
struct CustomerRecord
{
   enum {MAX_LENGTH = 32}; // how a constant gets tucked into a struct
   char Name[MAX_LENGTH];
   long Number;
   float BalanceDue;
};                          //note required semicolon after brace
```

In C++, the struct declaration above is a type definition and we can have declarations such as:

```
CustomerRecord  Cust,           // a single struct
                Customer[100];  // an array of structs

void Initialize (CustomerRecord & X); //prototype that uses a struct
```

The dot operator is used to reference the **fields** of a struct. Given the variable declarations above, the following code would be legal. (But don't be concerned with the logic!)

```
cin.getline(Cust.Name, Cust.MAX_LENGTH);//Cust.Name is string var
Cust.Number = 1234567;
Cust.BalanceDue = 0.0;
Customer[1] = Cust;        //copies all fields in the struct Cust
cout << Customer[1].Number;
```

Structs are enormously valuable tools in organizing program data. Programmers often place related variables in a struct, and then can pass instances of the struct to functions as a single entity. For example, the following struct would be useful in writing a program to keep track of the frequency of occurrence of words in a file.

```
struct WordRecord
{
   enum {MAX_WORD = 16};         //max number of chars in word
   char Letter[MAX_WORD + 1];    //the actual letters in the word
   unsigned Frequency;           //the occurrence of the word
};
```