1 Structs

Structs let you encapsulate one or more existing types into a new type. The classic example of a struct is a database record. If you are building a personnel systetm to keep track of employee information, you will need to store the first initial, last initial, employee number, and salary for each employee. A struct that contains all of this information is shown in the employeestruct.h header file that follows:

```
struct Employee {
    char firstInitial;
    char lastInitial;
    int employeeNumber;
    int salary;
};
```

A variable declared with type Employee will have all of these fields built-in. The individual fields of a struct can be accessed by using the ”.” operator. The example that follows creates and then outputs the record for an employee:

```
#include <iostream>
#include "employeestruct.h"
using namespace std;
int main()
{
    // Create and populate an employee.
    Employee anEmployee;
    anEmployee.firstInitial = 'M';
    anEmployee.lastInitial = 'G';
    anEmployee.employeeNumber = 42;
    anEmployee.salary = 80000;
    // Outpyut the values of an employee
    cout << "Employee: " << anEmployee.firstInitial  <<
                            anEmployee.lastInitial << endl;
    cout << "Number: " << anEmployee.employeeNumber << endl;
    cout << "Salary: $" << anEmployee.salary << endl;
    return 0;
}
```