3 Exercises

1. Here is a header file:

```
// golf.h
const int Len = 40;
struct golf
{
    char fullname[Len];
    int handicap;
};
// non-interactive version:
// function sets golf structure to provided name, handicap
// using values passed as arguments to the function
void setgolf(golf& g, const char* name, int hc);

// interactive version:
// function solicits name and handicap from user
// and sets the members of g to the values entered
// returns 1 if name is entered, 0 if name is empty string
int setgolf(golf& g);

// function resets handicap to new value
void handicap(golf& g, int hc);

// function displays contents of golf structure
void showgolf(const golf& g);
```

Note that setgolf() is overloaded. Using the first version of setgolf() would look like this:

golf ann;

setgolf(ann, "Ann Birdfree", 24);

The function call provides the information that's stored in the **ann** structure.

Using the second version of setgolf() would look like this:

golf andy;

setgolf(andy);

The function would prompt the user to enter the name and handicap and store them in the **andy** structure.

Put together a multifile program based on this header. One file, named **golf.cpp**, should provide suitable function definitions to match the prototypes in the header file. A second file named **main.cpp** should contain main() and demonstrate all the features of the prototyped functions. For example, a loop should solicit input for a name and handicap of the golf until the name is not empty. The main() function should use only the prototyped functions to access the golf structures.

A sample runs might look like this:

```
First version of setgolf function:
The name of golf is Ann Birdfree and its handicap is 24
Second version of setgolf function:
Enter the fullname:
Enter the handicap:2
Enter the fullname:
Enter the handicap:1
Enter the fullname:Andy Baffy
Enter the handicap:26
The name of golf is Andy Baffy and its handicap is 26
```

2. Write a three-file program based on the following namespace:

```
namespace SALES
{
    const int QUARTERS = 4;
    struct Sales
    {
        double sales[QUARTERS];
        double average;
        double max;
        double min;
    };

    // copies the lesser of 4 or n items from the array ar
    // to the sales member of s and computes and stores the
    // average, maximum, and minimum values of the entered items;
    // remaining elements of sales, if any, set to 0
    void setSales(Sales& s, const double ar[], int n);

    // gathers sales for 4 quarters interactively, stores them
    // in the sales member of s and computes and stores the
    // average, maximum, and minimum values
    void setSales(Sales& s);

    // display all information in structure s
    void showSales(const Sales& s);

}
```

The **first file** should be a header file that contains the namespace. The **second file** should be a source code file that extends the namespace to provide definitions for the three prototyped functions. The **third file** should declare two **Sales objects**. It should use the non-interactive version of setSales() to provide values for one structure and the interactive version of setSales() to provide values for the second structure. It should display the contents of both structures by using showSales().

A sample runs might look like this:

```
Non-interactive version of setSales() to provide values:
Sales:345.2   621.8   1073.5
Average:680.167
Max:1073.5
Min:345.2
Interactive version of setSales() to provide values:
Enter sales for 4 quarters: 459.3 902.1 1356.7 824.9
Sales:459.3   902.1   1356.7   824.9
Average:885.75
Max:1356.7
Min:459.3
```