Passing structs to Functions (7.5)

- structs can be passed to functions by reference or value in the same manner that other data types have been passed
- Generally, passing structs by reference is preferred since passing by value requires a local copy of the struct to be created within the function's variables

Example

```
struct Date
{
  int day,
      month,
      year;
};
```

- Create a date variable equal to February 9, 2009
- Write a function that accepts a Date and prints the date out in the form day-month-year

Nested Structures (7.4)

- Structures can be nested so that a member of a structure can be another structure

```
struct Friend
{
  string name;
  Date sBirthday;
};
```

- Write the code that will ask the user for a name and date and store that in a Friend struct variable

Arrays of structs (7.13)

- It is possible to declare an array of structs
- A datafile called athletes.txt exists which contains an unknown amount of information where each line of the file contains an id, age, and weight of a specific athlete. The program will contain two functions:
  - `void readAthleteData` - This function reads in up to 100 lines of data into an array of structs and returns the number of athletes in the datafile.
  - `int whatAge` - This function returns the age of the athlete with the given idNumber.
- Declare a struct for each athlete's data
- Create an array of structs to hold all athlete's data
- Write each function described above