**Instructions:**

1. Create a MS VisualStudio solution named **Assignment18**.
2. Create projects according to the assigned problems.
3. Use techniques and experience that you have obtained from previous chapters and this chapter.
4. Code and test your programs. They **MUST** be completed without syntax, logic and run-time errors.
5. Include comments of your name, date, and brief descriptions in all source codes.
6. Compress **Assignment18**folder into **ONE** zipped file.
7. Submit or re-submit your zipped file before its due date&time.

int main()  
{  
   do  
   {  
       switch (menuOption())  
       {  
       case 0: exit(1); break;  
     case 1: Challenge1(); break; //done  
      case 2: Challenge2(); break; //done  
      case 3: Challenge3(); break;  
     case 7: Challenge7(); break;  
      case 8: Challenge8(); break;  
      case 12: Challenge12(); break; // 20 points extra credit  
       default: cout << "\t\tERROR - Invalid option. Please re-enter."; break;  
       }  
       cout << "\n";  
       system("pause");  
  
    } while (true);  
  
   return EXIT\_SUCCESS;  
}

1. Your Own Linked List

Design your own linked list class to hold a series of integers. The class should have member functions for appending, inserting, and deleting nodes. Don’t forget to add a destructor that destroys the list. Demonstrate the class with a driver program.

1. List Print

Modify the linked list class you created in Programming Challenge 1 to add a print member function. The function should display all the values in the linked list. Test the class by starting with an empty list, adding some elements, then printing the resulting list out.

1. List Copy Constructor

Modify your linked list class of Programming Challenges 1 and 2 to add a copy constructor. Test your class by making a list, making a copy of the list, then displaying the values in the copy.

1. Member Removal by Position

Modify the list class you created in the previous programming challenges by adding a member function for deleting a node at a specified position. A value of 0 for the position means the first node in the list (the current head) is deleted. The function does nothing if the specified position is greater than or equal to the length of the list.

1. List Template

Create a list class template based on the list class you created in the previous programming challenges.

12. Double Merge

Modify the NumberList class shown in this chapter to include a member function named mergeArray. The mergeArray function should take an array of doubles as its first argument, and an integer as its second argument. (The second argument will specify the size of the array being passed into the first argument.)

The function should merge the values in the array into the linked list. The value in each element of the array should be inserted (not appended) into the linked list. When the values are inserted, they should be in numerical order. Demonstrate the function with a driver program. When you are satisfied with the function, incorporate it into the LinkedList template.