CSC 2000
Introduction to C++ Programming Language
Winter Term 2020
Project 01
50 points
Due 03/18/2020 (11:45 A.M.)

The objectives of this Project are:

1. Being able to Analyze, Design, implement, and test a practical real-world application.
2. Being able to use selections and repetition structures
3. Being able to deal with Functions
4. Being able to read and write to files.

Requirements:

In a word file:

- Analyze each problem; outline the problem and its solution requirements. (Describe the problem including input and output in your own words.))
- Design an algorithm to solve the problem. (Describe the major steps for solving the problem.)
- Using visual Studio C++ 2019 software, implement the algorithm in C++.
- Test the code for each problem and verify that the algorithm works; include a screenshot for program each program's output.

Restrictions:

You must work individually. Use only material from class or from the text book (chapters 1- 6). All code must be the work of the individual. Do not share your code or copy from external resources.

Submission

Submit two files for each problem, one-word file (for analysis, design, and screenshot) and the source file.
Upload all file to Canvas by the due date. DO NOT Email your files.

Grading Criteria:

The grade of each program will be based on the creation of a program that works correctly, up to some details (50%), clear problem analysis and algorithm design (5%), the appropriate use of functions and loops (25%), the production of clear output, with readable formatting and without unnecessary repetition (10%), composition of informative comments (5%), and testing the program with different inputs (5%). Programs must compile.

Problem 01: Fitness Center (30 points)

The cost to become a member of a fitness center is as follows:
(a) the senior citizens discount is 30%;
(b) if the membership is bought and paid for 12 or more months, the discount is 15%;
(c) if more than five personal training sessions are bought and paid for, the discount on each session is 20%.
Write a menu-driven program that determines the cost of a new membership. Your program must contain

1. a **function** that displays the general information about the fitness center and its charges.
2. a **function** to get all the necessary information to determine the membership cost.
3. a **function** to determine the membership cost.

Use appropriate parameters to pass information in and out of a function. (Do not use any global variables.)

Sample Output 1

```
Welcome to Stay Healty and Fit center.
This program determines the cost of a new membership.
If you are a senior citizen, then the discount is 30% of of the regular membership price.
If you buy membership for twelve months and pay today, the discount is 15%.
If you buy and pay for 6 or more personal training session today, the discount on each session is 20%.
Enter the cost of a regular membership per month: 55.50

Enter the cost of one personal training session: 15.0

Are you a senior citizen (Y,y/N,n): n

Enter the number of personal training sessions bought: 3

Enter the number of months you are paying for: 6

The membership cost = $378.00
```

Sample Output 2

```
Welcome to Stay Healty and Fit center.
This program determines the cost of a new membership.
If you are a senior citizen, then the discount is 30% of of the regular membership price.
If you buy membership for twelve months and pay today, the discount is 15%.
If you buy and pay for 6 or more personal training session today, the discount on each session is 20%.
Enter the cost of a regular membership per month: 55.50

Enter the cost of one personal training session: 15.0

Are you a senior citizen (Y,y/N,n): y

Enter the number of personal training sessions bought: 8

Enter the number of months you are paying for: 12

The membership cost = $492.27
```

Problem 02: Merge Sort (20 points)

Write a program that merges the numbers in two files and writes all the numbers into a third file. Your program takes input from two different files and writes its output to a third file. Each input file contains a list of numbers of type int in sorted order from the smallest to the largest. After the program is run, the output file will contain all the numbers in the two input files in one longer list in sorted order from smallest to largest. Your program should define at least one **function** that is called with the two input-file streams and the output-file stream as three arguments.
The list of numbers are not necessary of the same length. Aske the user for each file name and test if the file exit. Show the content of all three files on a screen.

Sample Input file 1:

22
44
55
77
99
500

Sample Input file 2:

11
55
101

Sample Output in a file:

11
22
44
55
55
77
99
101
500

Sample Output on a screen:

```
Enter the first of two input file names:
inName1.txt
Now a second input file name
inName2.txt
Enter the output file name.
WARNING: ANY EXISTING FILE WITH THIS NAME WILL BE ERASED.
outName.txt
Contents of file inName1.txt are:
22
44
55
77
99
500
Contents of file inName2.txt are:
11
55
101
Contents of merged file, outName.txt are:
11
22
44
55
55
77
99
101
500
```