Contestant Number: _____

Time: _____

Rank: _____

C++ PROGRAMMING
(335)

NATIONAL – 2019

Production Portion:

Job 1: Golden Oaks _____ (380 points)

TOTAL POINTS _____ ***(380 points)***

Failure to adhere to any of the following rules will result in disqualification:

1. **Contestant must hand in this test booklet and all printouts. Failure to do so will result in disqualification.**
2. **No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests or facsimile (handwritten, photocopied, or keyed) are allowed in the testing area.**
3. **Electronic devices will be monitored according to ACT standards.**

No more than ten (10) minutes orientation
No more than (90) minutes testing time
No more than ten (10) minutes wrap-up

Your application will be graded on the following criteria:

Solution and Project

Project was found on the flash drive	____ 10 points
Project is named following the naming convention	____ 10 points

Program Execution

Program runs	____ 20 points

If program does not execute, then remaining items in this section are not scored.

The program gracefully handles an empty or missing input file	____ 10 points
The program gracefully handles any data errors	____ 20 points
The program shows a course/member list/exit menu to choose from	____ 20 points
The program displays the complete list of members	____ 10 points
The program displays the sorted list of member's handicaps for each of the courses from largest to smallest	____ 30 points
The program displays the data to the user in clean even rows	____ 10 points
The program displays the course name above the members list	____ 5 points
The program displays a prompt for collecting user input and to return the user back to the menu.	____ 10 points

Source Code Review

Contestant ID is commented at the top of the program	____ 5 points
Code is commented at the top, for each method and as needed	____ 15 points
Code uses structs	____ 15 points
Code uses vectors(arrays) with structs(or classes)	____ 40 points
A method called "Calc_Course_Handicap" is implemented	____ 10 points
Calc_Course_Handicap code is commented to show what is being used to calculate the MGS.	____ 5 points
Code uses reasonable and consistent variable/method/struct naming conventions	____ 15 points
Code uses a sort algorithm to sort handicaps for each course	____ 40 points
The program reads the data into the correct data structure(s) from a file called "Golf Data.txt"	____ 40 points
The program gracefully handles remaining error checking	____ 20 points
Code has a minimum of one pass by reference usage in a method	____ 10 points
The program includes code to prompt the use to continue when they are ready.	____ 10 points

Total ____/380 points

Golden Oaks

The local country club wants you to build them a golf handicap program that members can use to find other members to play golf rounds with – for all six of their courses.
Golden Oaks Supplied:

1. Data file called "Golf Data.txt" which has all the related program data.
2. Data file format:
    a. Course Name, Rating, and Slope – six (6) times.
    b. Member's –
        i. Name
        ii. Club ID#
        iii. # of nine (9) hole rounds followed by that many scores for that course
        iv. # of eighteen (18) hole rounds followed by that many scores for that course
            - These two steps are repeated six(6) times total for the six(6) courses
    c. Number of rounds will always have values from zero (0) to twenty (20) scores.
    d. A round's score will always be less than two hundred (200).
3. To calculate a player's course handicap:
    a. Calculate a Modified Gross Score(MGS) by one of these ways:
        i. Total 5 to 20 rounds of eighteen (18) holes.
        ii. Total 10 to 20 rounds of nine (9) holes.
        iii. Total of nine(9) holes < 10 and eighteen(18) holes <10 take all the nine (9) hole rounds and double the scores, then add all the 18 hole scores.
    b. Take an average to find the

$$\text{Modifide Gross Score} = \frac{\text{Sum of Rounds}}{\text{Number of Rounds}}$$

MGS
Fig#1
    c. Take the MGS and subtract the course rating.
    d. Take that value and multiply it by 113.
    e. Divide that value by course's slope rating. The answer is the player's Handicap Index (HI).

$$\text{Handicap Index} = \frac{(\text{MGS} - \text{Course Rating}) \times 113}{\text{Slope Rating}}$$

Fig#2

f. Take the HI and multiply it by 0.96
g. Take the answer and multiply that by the slope rating.
h. Then divide by 113 rounding to the nearest whole number.

$$\text{Course Handicap} = \frac{\text{HI X 0.96 X Slope Rating}}{113}$$

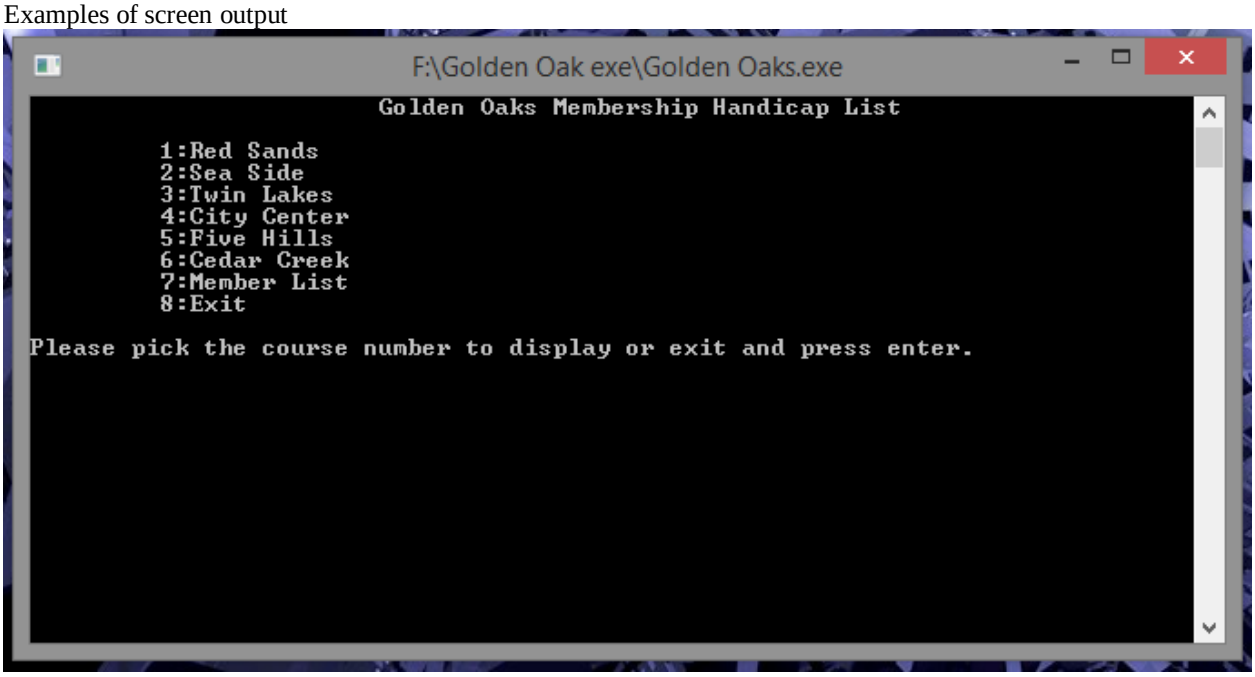
Round to the nearest whole number **Fig#3**

4. Provide a menu so the user can pick from the course, member list or to exit the program.
5. Display a sorted list of members for any course the user chooses – largest to smallest. (Image 3)

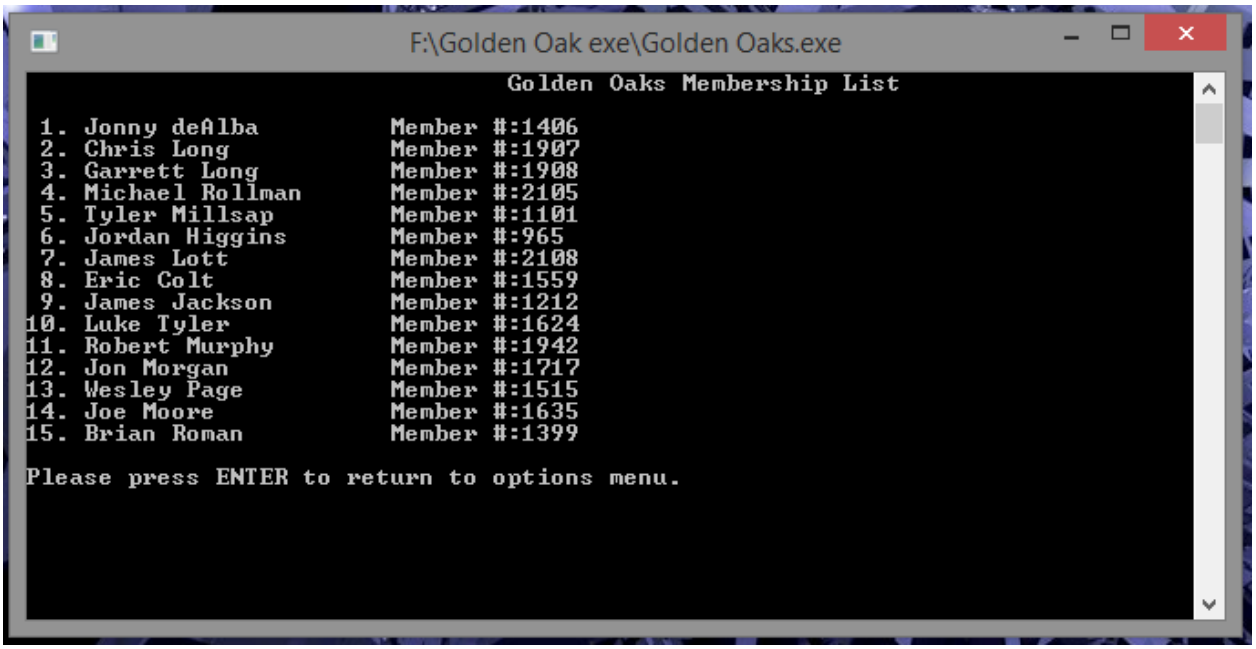
Developer inclusions:

1. Use structs or classes in your code.
2. Vectors (arrays)
3. Method call Calc_Course_Handicap
4. Sort Algorithm
5. Clear the display between screen changes.
6. Comments for methods and important code segments.
7. ID number as a comment at the top of the code.
8. Logical names for constants, variables and other methods
9. Use Pass by Reference in at least one method.
10. Code to wait for the user to continue when ready.
11. Displayed data is in clean even rows for ease of reading by user.

Examples of screen output



Image#1



Image#2

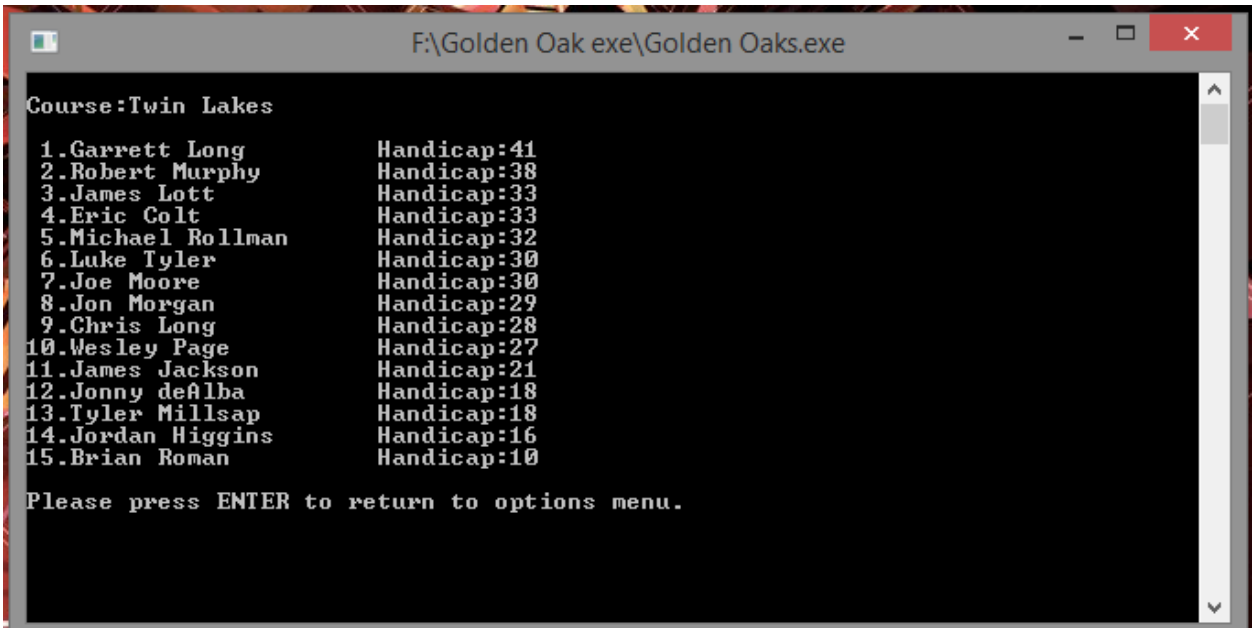


Image #3