



Course: CS1004 Object Oriented Programming.

Instructor: Lecturer Sara Rehmat.

Submitted by: Muhammad Rehan

Roll no: 22P-9106

Class: BSE-2A

Date: February 13th 2023.

Home Task 01

Department of Computer Science

Code:

```
// Muhammad Rehan | 22P-9106 | BSE-2A | HomeTask No 01. |
#include <iostream>
#include <string>
#include <string.h>
// Including this to use strtok() function.
#include <stdbool.h>
// Including this to use bool variables.
#define studentStrength 20
//Defining students strength
#define space "\n\n\n"
#define tab "\t"
// Defining these so i can use this in program instead of writing long \n and \ts.
using namespace std;
// Creating Structure blueprint.
struct studentInfoType
{
    string studentFName, studentLName;
    int testScore;
    char grade;
};
//Functions Declartion.
void FLUSH();
//Function to clean the input buffer or stream.
void Operations(void);
//Function which will operate the whole program.
void infoEntery(studentInfoType Students[]);
// Function to do entry.
void gradeAssigning(studentInfoType Students[]);
// Function to assign grade according to the testscores.
bool DisplayData(studentInfoType Students[], studentInfoType AscendArray[],
studentInfoType DescendArray[]);
// Function to display data.
void sortedArray(studentInfoType Students[], studentInfoType AscendArray[],
studentInfoType DescendArray[]);
// Function to Sort the arrays ascending and descending way.
void Display(studentInfoType *Data);
// A Function to display the sheet.
void highestMarks(studentInfoType *Data);
// A Function to display highest marks.
void highestMarksStudents(studentInfoType *Data);
// A Function to display highest marks student.
int main()
{
    //Starting the program.
    Operations();
}
void Operations(void)
{
    // Creating arrays of the structs
    studentInfoType Students[studentStrength];
```

```

studentInfoType AscendArray[studentStrength];
studentInfoType DescendArray[studentStrength];
int option;
// calling data entry function
infoEntery(Students);
// Assigning the grades
gradeAssigning(Students);
// Sorting our arrays
sortedArray(Students, AscendArray, DescendArray);
// this loop will run till our user enters 4 as input.
while (true)
{
    int option;
    cout << "Menu:\n1. Display Student Data\n2. Find Highest Test score\n3.
Print Students having the highest test score.\n4. Exit\n=>", cin >> option;
    while (true)
    {
        switch (option)
        {
            case 1:
                // If this function returns false this function will keep calling
                // if the function return true the loop will break.
                while (true)
                {
                    if (DisplayData(Students, AscendArray, DescendArray))
                    {
                        break;
                    }
                    break;
                }
            case 2:
                // Calling the funtion to do its job.
                highestMarks(DescendArray);
                break;
            case 3:
                // Calling the funtion to do its job.
                highestMarksStudents(DescendArray);
                break;
            case 4:
                return;
                break;
            default:
                cout << "\nEnter Valid Input.\n";
                break;
        }
        break;
    }
}
}

void infoEntery(studentInfoType Students[])
{
    // So in this function i dont want user to enter the first and last name to
    enter differently. so make it easy for user to enter the full name and then
    split it into the first and last name. make things easy for user.
    char fullName[100];

```

```

// this will store the fullname
cout << "-:Kindly Enter the Students Information:-" << endl;
for (int i = 0; i < studentStrength; i++)
{
    cout << tab << "-> Kindly the enter First Name followed by Last name
studentno " << i + 1 << " : ", cin.getline(fullName, 100), cout << tab << "->
Kindly enter the testscore of " << fullName << " : ", cin >>
Students[i].testScore;
    // The following code will split the names into first and last.
    char *name = strtok(fullName, " ");
    // Space will be the break point.
    Students[i].studentFName = name;
    name = strtok(NULL, "\n");
    Students[i].studentLName = name;
    FLUSH();
}
}
void gradeAssigning(studentInfoType Students[])
{
    // using a different approach and using ternary operator instead of if and
    else.
    for (int i = 0; i < studentStrength; i++)
    {
        Students[i].testScore > 90 ? Students[i].testScore = 'A' :
Students[i].testScore < 90 && Students[i].testScore > 80 ? Students[i].grade =
'B'
:
Students[i].testScore < 80 && Students[i].testScore > 70 ? Students[i].grade
= 'C'
:
Students[i].testScore < 70 && Students[i].testScore > 60 ? Students[i].grade
= 'D'
: Students[i].grade = 'F';
    }
}
bool DisplayData(studentInfoType Students[], studentInfoType AscendArray[],
studentInfoType DescendArray[])
{
    // Use code to present user the option to display the data into three
    different ways.
    int option;
    cout << "Press:\n1. For Ascending Order\n2. Descending order.\n3. For
Number Wise\n=>", cin >> option;
    switch (option)
    {
        case 1:
        {
            Display(AscendArray);
            return true;
        }
        break;
        case 2:
        {

```

```

        Display(DescendArray);
        return true;
    }
    break;
    case 3:
    {
        Display(Students);
        return true;
    }
    break;
    default:
    {
        cout << "Enter a valid input" << endl;
        return false;
    }
    break;
}
}

void sortedArray(studentInfoType Students[], studentInfoType AscendArray[],
studentInfoType DescendArray[])
{
    // This code will first make copy in Student array into AscendArray and
    DescendArray.
    for (int i = 0; i < studentStrength; i++)
    {
        DescendArray[i] = AscendArray[i] = Students[i];
    }
    // Now sorting the arrays using bubble sort Ascending wise.
    for (int i = 0; i < studentStrength - 1; i++)
    {
        for (int j = 0; j < studentStrength - i - 1; j++)
        {
            if (AscendArray[j].testScore > AscendArray[j + 1].testScore)
            {
                studentInfoType temp = AscendArray[j];
                AscendArray[j] = AscendArray[j + 1];
                AscendArray[j + 1] = temp;
            }
        }
    }
    // Now sorting the arrays using bubble sort Descending wise.
    for (int i = 0; i < studentStrength - 1; i++)
    {
        for (int j = 0; j < studentStrength - i - 1; j++)
        {
            if (DescendArray[j].testScore < DescendArray[j + 1].testScore)
            {
                studentInfoType temp = DescendArray[j];
                DescendArray[j] = DescendArray[j + 1];
                DescendArray[j + 1] = temp;
            }
        }
    }
}
}

```

```

void Display(studentInfoType *Data)
{
    // Displaying the data sheet the way instructed. and it will be left
    justified by default.
    cout << endl;
    for (int i = 0; i < studentStrength; i++)
    {
        cout << Data[i].studentLName << ", " << Data[i].studentFName << ", " <<
Data[i].testScore << ", " << Data[i].grade << endl;
    }
    cout << endl;
}
void highestMarks(studentInfoType *Data)
{
    // Showing the highest marks.
    int option;
    cout << "Kindly Enter the postion for which to want to print?:", cin >>
option;
    cout << endl;
    for (int i = 0; i < option; i++)
    {
        cout << Data[i].testScore << endl;
    }
    cout << endl;
}
void highestMarksStudents(studentInfoType *Data)
{
    // Showing the highest marks students.
    int option;
    cout << "Kindly Enter the postion for which to want to print?:", cin >>
option;
    cout << endl;
    for (int i = 0; i < option; i++)
    {
        cout << Data[i].studentFName << " " << Data[i].studentLName << endl;
    }
    cout << endl;
}
void FLUSH()
{
    // Clearing the input stream.
    while (getchar() != '\n')
    {
    }
}

```