**Fundamentals of Programming**

**Open Ended Lab**

**Report for Grade Management System**

Submitted by: Samrah Mumtaz

Course: BESE 29-C

Submission Date: 21st January 2024

Submitted to: Sir Asif

Introduction:

The Grade Management System is a C++ program designed to input and manage grades for a specified number of students and subjects. It allows users to input grades, view the grades in a tabular format, and update grades if necessary.

Key Features:

1. Input and Error Handling:

The “takeInput” function is responsible for taking input from the user for each student and subject, incorporating error handling to ensure valid input (grades between 0 and 100).

1. Grading System:

The “toGrades” function translates numeric grades into letter grades (A, B, C, D, F).

1. Printing Grades in Table Format:

The “printGrades” function displays the grades of all students in a table format, including student indices, subject indices, and corresponding grades.

1. Calculating Averages:

The “calculateAverages” function calculates student-wise and subject-wise averages based on the entered grades.

1. Updating Grades:

The “updateMarks” function allows users to update the grades for a specific student and subject, ensuring input validation to maintain data integrity.

1. Interactive User Interface:

The program provides an interactive interface, allowing users to continuously update grades and view the results.

Improvements Made:

1. Enhanced Error Handling:

Improved error handling in the updateMarks function to validate the input grades, preventing invalid entries.

1. Clear Screen Function:

Used system("cls") to provide with clearer output.

1. Structured Output:

Enhanced the structure of the displayed tables for better readability and clarity.

1. Dynamic Array Sizes:

Allows dynamic sizes for the array based on user input for more flexibility.

Code Outline:

#include <conio.h>

#include <iostream>

#include <iomanip>

#include <windows.h>

using namespace std;

// Function for taking input and error handling

void takeInput ()

// Function for marks to grades

char toGrades (int marks)

// Funciton to print all grades

void printGrades ()

// Function for student averages and subject averages

void calculateAverages(char studentGrades[], char subjectAverages[])

// Function to display student-wise and subject-wise grades along with their averages

void printResult (const char studentGrades[], const char subjectAverages[])

// Function for updating marks

void updateMarks ()

int main()

Library Usage:

**iostream:** for basic input output functions

**conio.h:** for function getch() and getchar()

**iomanip:** for setw() function

**windows.h:** for system(“cls”) function to clear screen

Conclusion and Recommendation:

The Grade Management System provides a functional solution for managing student grades with an intuitive user interface. The implemented improvements enhance the overall robustness and user experience. Further enhancements and refinements could be considered based on specific user requirements and future development plans. For example:

1. Modularization:

Further modularize the code to improve readability and maintainability. Break down functions into smaller, single-responsibility functions. For example, create separate functions for input, grading, printing, and updating.

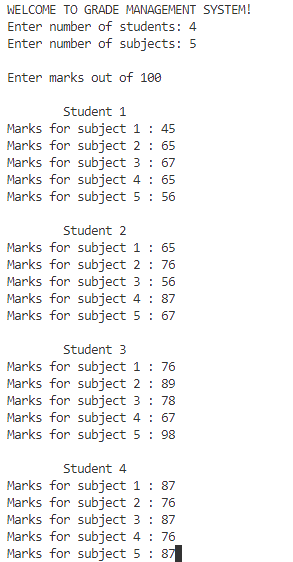
1. Encapsulation:

Encapsulate related functionalities into classes if the program grows in complexity. This can lead to more maintainable and scalable code.

1. Dynamic Memory Allocation:

Instead of using fixed-size arrays, you can allocate memory dynamically based on user input. Use **new** and **delete** operators to allocate memory for the marks array.

Code Snippets:

 A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Code:

#include <conio.h>

#include <iostream>

#include <iomanip>

#include <windows.h>

using namespace std;

// Global Variables

int students, subjects;

int marks[3][5];

// Function for taking input and error handling

void takeInput ()

{

    for (int  i = 0; i < students; i++)

    {

        cout << "\n\tStudent " << (i + 1) << endl;

        for (int j = 0; j < subjects; j++)

        {

            // For error handling

            do

            {

                cout << "Marks for subject " << (j + 1) << " : ";

                cin >> marks[i][j];

            }

            while (marks[i][j] < 0 || marks[i][j] > 100);

        }

    }

}

// Function for marks to grades

char toGrades (int marks)

{

    if (marks >= 90)

        return 'A';

    else if (marks >= 80)

        return 'B';

    else if (marks >= 70)

        return 'C';

    else if (marks >= 60)

        return 'D';

    else

        return 'F';

}

// Funciton to print all grades

void printGrades ()

{

    cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

    cout << "|" << setw(10) << "  Students\t";

    for (int k = 0; k < subjects; k++)

    {

        cout << "|" << setw(8) << "   Subject " << (k + 1) << "\t";

    }

    cout << "|" << endl;

    cout << "|---------------+---------------+---------------+---------------+---------------+---------------|"<<endl;

    for (int i = 0; i < students; i++)

    {

        cout << "|" << setw(10) << "Student " << (i + 1) << "\t";

        for (int j = 0; j < subjects; j++)

        {

            cout << "|" << setw(8) << toGrades(marks[i][j]) << "\t";

        }

        cout << "|" << endl;

    }

    cout << " ----------------------------------------------------------------------------------------------- "<<endl;

}

// Function for student averages and subject averages

void calculateAverages(char studentGrades[], char subjectAverages[])

{

    for (int i = 0; i < students; ++i)

    {

        int sum = 0;

        for (int j = 0; j < subjects; ++j)

        {

            sum += marks[i][j];

        }

        studentGrades[i] = toGrades(sum / subjects);

    }

    for (int j = 0; j < subjects; ++j)

    {

        int sum = 0;

        for (int i = 0; i < students; ++i)

        {

            sum += marks[i][j];

        }

        subjectAverages[j] = toGrades(sum / students);

    }

}

// Function to display student-wise and subject-wise grades along with their averages

void printResult (const char studentGrades[], const char subjectAverages[])

{

    // Print Student-wise Average Grades Table

    cout << "\nStudent-wise Average Grades:\n";

    cout << "+----------------------+-------+\n";

    cout << "| Student              | Grade |\n";

    cout << "+----------------------+-------+\n";

    for (int i = 0; i < students; ++i)

    {

        cout << "| Student " << setw(2) << (i + 1) << "           |   " << studentGrades[i] << "   |\n";

    }

    cout << "+----------------------+-------+\n";

    // Print Subject-wise Average Grades Table

    cout << "\nSubject-wise Average Grades:\n";

    cout << "+----------------------+-------+\n";

    cout << "| Subject              | Grade |\n";

    cout << "+----------------------+-------+\n";

    for (int j = 0; j < subjects; ++j)

    {

        cout << "| Subject " << setw(2) << (j + 1) << "           |   " << subjectAverages[j] << "   |\n";

    }

    cout << "+----------------------+-------+\n";

}

// Function for updating marks

void updateMarks ()

{

    int studentIndex, subjectIndex;

    cout << "Enter the student index (1-" << students << ") to update grades: ";

    cin >> studentIndex;

    // Error Handling

    if (studentIndex < 1 || studentIndex > students)

    {

        cout << "Invalid student index.\n";

        return;

    }

    cout << "Enter the subject index (1-" << subjects << ") to update grades: ";

    cin >> subjectIndex;

    // Error Handling

    if (subjectIndex < 1 || subjectIndex > subjects)

    {

        cout << "Invalid subject index.\n";

        return;

    }

    do

    {

        cout << "Enter new marks for Student " << studentIndex << ", Subject " << subjectIndex << ": ";

        cin >> marks[studentIndex - 1][subjectIndex - 1];

    }

    // Error Handling

    while (marks[studentIndex - 1][subjectIndex - 1] > 100);

}

int main()

{

    cout << "WELCOME TO GRADE MANAGEMENT SYSTEM!" << endl;

    cout << "Enter number of students: " ;

    cin >> students;

    cout << "Enter number of subjects: ";

    cin >> subjects;

    cout << "\nEnter marks out of 100 "<< endl;

    char studentGrades[students];

    char subjectAverages[subjects];

    // Task 1: Input marks for each student

    takeInput();

    // Task 2: Classify each student's marks into categories

    // Task 3: Calculate average grade for each student and each subject

    calculateAverages(studentGrades, subjectAverages);

    system("cls");

    cout << "WELCOME TO GRADE MANAGEMENT SYSTEM!" << endl;

    //Task 4: Display all grades

    printGrades();

    // Task 5: Display student-wise and subject-wise grades along with averages

    printResult(studentGrades, subjectAverages);

    while (true)

    {

        // Task 6: Allow the user to update grades

        cout << "Do you want to update grades?(Y/N)\t";

        char option;

        cin >> option;

        if ((option) == 'y' || option == 'Y')

        {

            updateMarks();

            // Recalculate averages after the update

            calculateAverages(studentGrades, subjectAverages);

            // Print it anew

            system("cls");

            cout << "WELCOME TO GRADE MANAGEMENT SYSTEM!" << endl;

            cout << "UPDATED GRADES: "<< endl;

            printGrades();

            printResult(studentGrades, subjectAverages);

        }

        else

        {

            break; // Exit the loop if the user chooses not to update marks

        }

    }

    getche();

    return 0;

}