Midterm Skills Exam 2025	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed:** 10/9/2025
Section: CPE11S1	**Date Submitted:** 10/9/2025
Name(s): Ramirez, Angel Mae C.	**Instructor:** Engr. Jimlord M. Quejado

6. Output

1. Create a program that records the grades of several students and computes their average grades. Use structures to store each student's data, arrays to handle multiple students, and loops for data input and processing.

CODE:

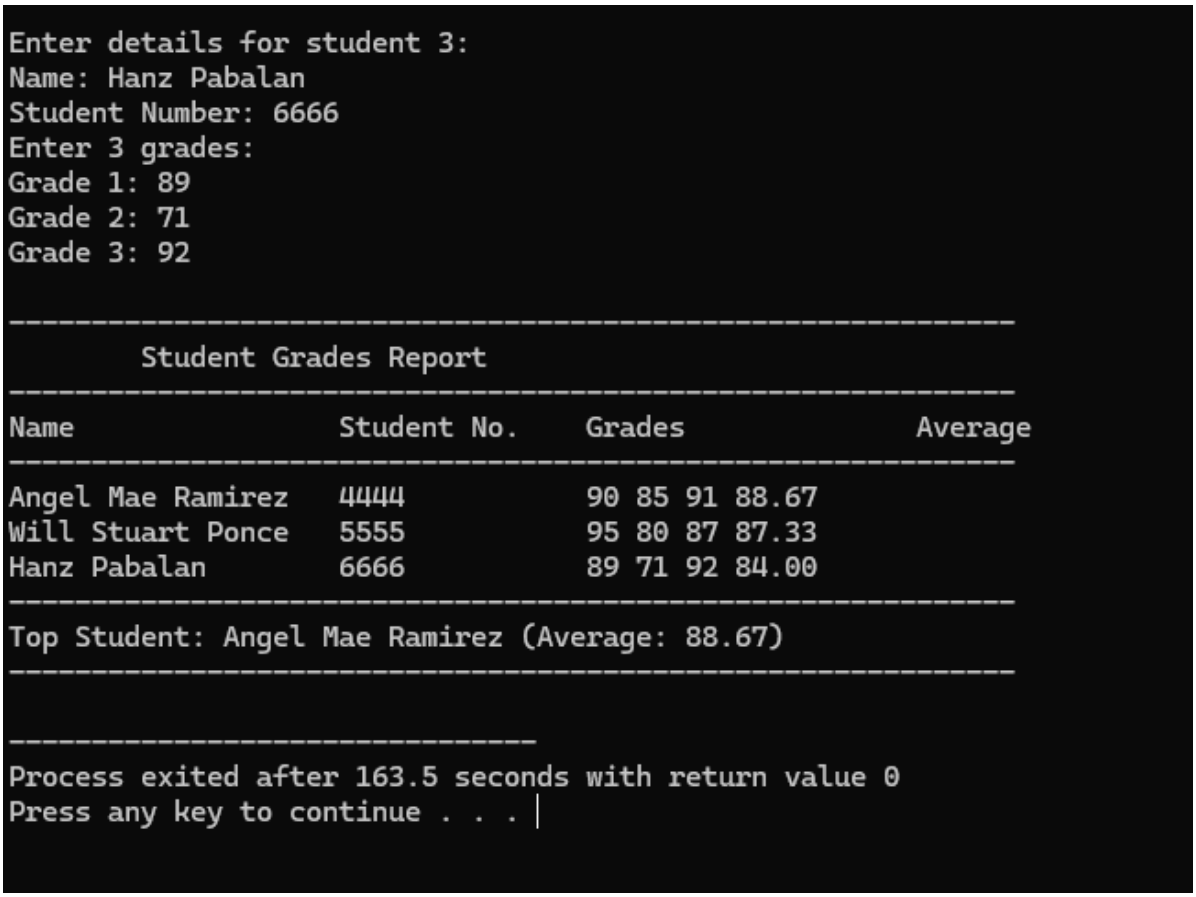
Ramirez_AngelMae_MidtermsSkillsExam.cpp

```cpp
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;


struct Student {
    string name;
    string studentNumber;
    double grades[3];
    double average;
};

int main() {
    Student students[10];
    int numStudents;

    cout << "Enter number of students (max 10): ";
    cin >> numStudents;


    if (numStudents > 10 || numStudents < 1) {
        cout << "Invalid number of students. Please enter between 1 and 10.";
        return 0;
    }

    // STUDENT DETAILS HERE
    for (int i = 0; i < numStudents; i++) {
        cout << "\nEnter details for student " << i + 1 << ":\n";
        cin.ignore();
        cout << "Name: ";
        getline(cin, students[i].name);
        cout << "Student Number: ";
        getline(cin, students[i].studentNumber);

        cout << "Enter 3 grades:\n";
        double sum = 0;
        for (int j = 0; j < 3; j++) {
```

```
            cout << "Grade " << j + 1 << ": ";
            cin >> students[i].grades[j];
            sum += students[i].grades[j];
        }
        students[i].average = sum / 3;
    }


    cout << "\n-----------------------------------------------------------\n";
    cout << setw(30) << "STUDENT GRADE REPORT\n";
    cout << "-----------------------------------------------------------\n";
    cout << left << setw(20) << "Name"
         << setw(15) << "Student No."
         << setw(20) << "Grades"
         << "Average\n";
    cout << "-----------------------------------------------------------\n";


    double highestAvg = 0;
    string topStudent;
    for (int i = 0; i < numStudents; i++) {
        cout << left << setw(20) << students[i].name
             << setw(15) << students[i].studentNumber;

        for (int j = 0; j < 3; j++) {
            cout << fixed << setprecision(0) << students[i].grades[j] << " ";
        }

        cout << setw(10) << fixed << setprecision(2) << students[i].average << endl;

        if (students[i].average > highestAvg) {
            highestAvg = students[i].average;
            topStudent = students[i].name;
        }
    }

    cout << "-----------------------------------------------------------\n";
    cout << "Top Student: " << topStudent << " (Average: " << fixed << setprecision(2) << highestAvg << ")\n";
    cout << "-----------------------------------------------------------\n";

    return 0;
}
```

OUTPUT:

```
C:\Users\TIPQC\Documents\F
Enter number of students (max 10): 3

Enter details for student 1:
Name: Angel Mae Ramirez
Student Number: 4444
Enter 3 grades:
Grade 1: 90
Grade 2: 85
Grade 3: 91

Enter details for student 2:
Name: Will Stuart Ponce
Student Number: 5555
Enter 3 grades:
Grade 1: 95
Grade 2: 80
Grade 3: 87
```



8. Conclusion

In this midterm skills test we were tasked to create a program called Student Average Grade Calculator that uses structures, arrays, and loops to record and compute the average grades of several students. Through this program, I learned how to organize data efficiently by using a structure, which allowed me to group each student's name, student number, and grades together. I also used arrays to handle multiple students at once, making it easier to store and process their information. By applying loops, I am able to repeatedly ask for inputs, calculate each student's average, and display all the results neatly in a summary table. I also implemented a condition to find and display the student with the highest average, which helped me practice using comparison and logic in programming. Overall, this activity helped me understand how structures, arrays, and loops work together to solve problems efficiently. It strengthened my programming skills, especially in data handling and presentation even though my code was full of errors and I nearly gave up a thousand times. I realized that even a simple program like this can make calculations faster. And I would definitely practice this when i get home.