Student Grades Tracker Program Documentation

Project Title: Tracker for Student Grades

This program computes the average grade of students by letting users enter the students' names and their corresponding grades.

Features:

- Input and display of student names and grades.

- Calculation of average grades.

- Error handling for invalid inputs.

- Option to add or remove students from the list.

C Program Code Example:

```c
#include <stdio.h>

 #include <string.h>

#define MAX_STUDENTS 100

#define NAME_LENGTH 50


typedef struct {

char name[NAME_LENGTH]; float grade;

} Student;

Student students[MAX_STUDENTS]; int student_count = 0;

void add_student() {

if (student_count >= MAX_STUDENTS) { printf("Student list is full.\n"); return;

}

char name[NAME_LENGTH]; float grade;

printf("Enter student's name: "); scanf("%s", name);

printf("Enter student's grade (0-100): ");

if (scanf("%f", &grade) != 1 || grade < 0 || grade > 100) { printf("Invalid
grade. Please enter a number between 0 and 100.\n"); while (getchar() != '\n');
// Clear the input buffer

return;

}

strcpy(students[student_count].name, name); students[student_count].grade =
grade; student_count++;

printf("%s added with grade %.2f.\n", name, grade);
```

```c
}
void remove_student() {
if (student_count == 0) {
printf("No students to remove.\n"); return;
}
char name[NAME_LENGTH];
printf("Enter the name of the student to remove: "); scanf("%s", name);
int found = 0;
for (int i = 0; i < student_count; i++) {
if (strcmp(students[i].name, name) == 0) {
for (int j = i; j < student_count - 1; j++) { students[j] = students[j + 1];
}
student_count--; found = 1;
printf("%s has been removed.\n", name); break;
}
}
if (!found) {
printf("Student not found.\n");
}
}
void display_grades() {
if (student_count == 0) {
printf("No students to display.\n"); return;
}
printf("\nStudent Grades:\n");
for (int i = 0; i < student_count; i++) {
printf("%s: %.2f\n", students[i].name, students[i].grade);
}
}
void calculate_average() {
if (student_count == 0) {
printf("No grades available to calculate an average.\n"); return;
}
float total = 0;
```

```c
for (int i = 0; i < student_count; i++) { total += students[i].grade;

}

float average = total / student_count; printf("\nAverage Grade: %.2f\n",
average);

}

int main() {

int choice;

do {

printf("\nGrade Tracker Menu:\n"); printf("1. Add Student\n"); printf("2. Remove
Student\n"); printf("3. Display Grades\n");

printf("4. Calculate Average Grade\n"); printf("5. Exit\n");

printf("Choose an option (1-5): ");

if (scanf("%d", &choice) != 1) {

printf("Invalid input. Please enter a number between 1 and 5.\n"); while
(getchar() != '\n'); // Clear the input buffer

continue;

}

switch (choice) { case 1:

add_student(); break;

case 2:

remove_student(); break;

case 3:

display_grades(); break;

case 4:

calculate_average(); break;

case 5:

printf("Exiting the Grade Tracker. Goodbye!\n"); break;

default:

printf("Invalid option. Please choose a number between 1 and 5.\n");

}

} while (choice != 5);

return 0;

}
```

Explanation of Features:

1. Adding Students:
   - Allows the user to add a student's name and grade, with validation to ensure the grade is between 0 and 100. If the input is invalid, the program clears the buffer and displays an error message.

2. Removing Students:
   - The user can remove a student by name. The function searches for the student, removes them, and shifts the remaining students in the list to maintain order.

3. Displaying Grades:
   - Lists all students and their grades. If there are no students, it notifies the user.

4. Calculating Average:
   - Computes and displays the average of all grades entered.

5. Error Handling:
   - Validates menu choices to prevent invalid or non-numeric input. For grade entries, it ensures values fall within the valid range of 0-100.

Output:
```
Grade Tracker Menu:
1. Add Student
2. Remove Student
3. Display Grades
4. Calculate Average Grade
5. Exit
Choose an option (1-5):
```