Project 4 Design Document

Program Requirements

In this project, we are tasked with developing a program that computes a weighted average for a student enrolled in a college course that is nearing the end of the semester; due to this, the four types of grades for the student should only add up to 80% total, as the final exam (not yet taken) will count for the other 20%. Exams count for 25% of the student’s grade, projects count for 25% of the student’s grade, assignments count for 15% of the student’s grade, and quizzes count for 15% of the student’s grade. The main body of the code has already been written, and we are only tasked with using the four following functions to complete our task: **void printPrompt(int a)**, **double average()**, **double weightedAverage(double, double, double, double)**,and **void outputAverages(double, double, double, double, double)**.

Program Inputs

There will be four different forms of input in this program. Each input will be an integer or a series of integers representing grades in each category.

* Quiz Grades
  + The quiz grades will be entered first as a series of integer values.
  + The integers entered in this category will be averaged and stored in **double quizAvg** for output and later use in finding the weighted average.
  + The quiz grades will count for 15% of the final weighted average.
* Assignment Grades
  + The assignment grades will be entered second as a series of integer values.
  + The values in this category will be averaged and stored in **double assignAvg** for output and later use in finding the weighted average.
  + The assignment grades will count for 15% of the final weighted average.
* Exam Grades
  + The exam grades will be entered third as a series of integer values.
  + The values in this category will be averaged and stored in **double examAvg** for output and later use in finding the weighted average.
  + The exam grades will count for 25% of the final weighted average.
* Project Grades
  + The project grades will be entered last as a series of integer values.
  + The values in this category will be averaged and stored in **double projAvg** for output and later use in finding the final weighted averaged.
  + The project grades will count for 25% of the final weighted average.

Program Outputs

The program will output 5 values through the function **void outputAverages(double, double, double, double, double)**. 5 double variables will be used as the parameters in the function.

* **double currentGrade**
  + This variable represents the student’s current grade and will be outputted first.
  + The value is found by calculating the weighted average of the student’s quiz, assignment, project, and exam grades. The quiz average comprises 15% of the grade, and the assignment average comprises another 15% of the grade. The exam average comprises 25% of the grade, and the project average comprises another 25% of the grade. This doesn’t take into account the final exam, however, so the percentages only add up to a total of 80%.
* **double quizAvg**
  + This variable represents the student’s quiz grades and will be outputted after the current grade.
  + The value is found by averaging the student’s quiz grades.
* **double assignAvg**
  + This variable represents the student’s assignment averages and will be outputted third, after the quiz average.
  + The value is found by averaging the student’s assignment grades.
* **double examAvg**
  + This variable represents the student’s exam averages and will be outputted fourth, after the assignment average.
  + The value is found by averaging the student’s exam grades.
* **double projAvg**
  + This variable represents the student’s project averages and will be outputted last, after the exam average.
  + The value is found by averaging the student’s project grades.

Test Plan

In order to ensure that the program is working correctly, I will establish a number of test cases with varying scenarios to compare the program’s results and verify that it’s working as intended. In the tables presented for each test case, I will be omitting the “-1” used within the actual input that signifies that input for that category is finished.

Case 1: The student has various grades within each category.

|  |  |  |
| --- | --- | --- |
| Category | Inputs | Outputs |
| Quiz Grades | 75, 63, 89, 95 | 80.5 |
| Assignment Grades | 80, 80, 85, 100 | 86.25 |
| Exam Grades | 88 | 88 |
| Project Grades | 80, 70 | 75 |
| Current Grade | ((80.5\*15)+(86.25\*15)+(88\*25)+(75\*25))  80 | 82.2031 |

Case 2: Although unlikely in practice, in this case, the student has the same grade for every quiz, assignment, exam, and project.

|  |  |  |
| --- | --- | --- |
| Category | Inputs | Outputs |
| Quiz Grades | 80, 80, 80, 80 | 80 |
| Assignment Grades | 80, 80, 80, 80, 80 | 80 |
| Exam Grades | 80, 80 | 80 |
| Project Grades | 80 | 80 |
| Current Grade | ((80\*15)+(80\*15)+(80\*25)+(80\*25))  80 | 80 |

Case 3: In this case, the student has elected to turn in no assignments.

|  |  |  |
| --- | --- | --- |
| Category | Inputs | Outputs |
| Quiz Grades | 89, 84, 76 | 83 |
| Assignment Grades | 0, 0, 0, 0, 0 | 0 |
| Exam Grades | 75, 82 | 78.5 |
| Project Grades | 71 | 71 |
| Current Grade | ((83\*15)+(78.5\*25)+(71\*25))  80 | 62.2813 |

Solution Overview

Since the main body of code is already provided for this program, only four other functions have to be defined and utilized. These functions are **void printPrompt(int a)**, **double average()**, **double weightedAverage(double, double, double, double)**,and **void outputAverages(double, double, double, double, double)**.

For **void printPrompt(int a)**, we must create a piece of code that relies on a switch statement in order to correctly prompt the user to enter the correct series of inputs. This switch statement will take the value of **int a** (which, in this case, is **count** within the main body of code) and accordingly select which prompt to output to the user. Within the main body of code, **count** goes up by one with each iteration of the while loop that **void printPrompt(int a)** is called in. Knowing this, we can create cases for the values of 1 through 4, as **count** is first assigned the value of 1, and the while loop will run until **count** is greater than 4. For case 1, **void printPrompt(int a)** will prompt the user to input quiz grades. For case 2, **void printPrompt(int a)** will prompt the user to input assignment grades. For case 3, **void printPrompt(int a)** will prompt the user to input exam grades. Finally, for case 4, **void printPrompt(int a)** will prompt the user to input project grades.

After being prompted to input grades, **double average()** will accept the inputs from the user. A switch statement within the main body of code will call the function, and from there, **double average()** will accept the first input from the user. The function will then use a sentinel-controlled while loop set to continuously run until the value inputted is “-1”. This while loop will add 1 to a count value, add the value inputted to a running sum, and then accept further input. After the user inputs a “-1,” **double average()** will divide the running sum by the count value in order to establish an average for the inputs. The function will then return this average, and the main function of the program will assign this average to the corresponding average value within the main body.

After all of the values have been averaged, **double weightedAverage(double, double, double, double)** will then be called. The various average previously generated by **double average()** will be entered into the function’s parameters. These values will be used to compute a weighted average, which is the student’s current grade in the class. In order to compute the weighted average, the values of each of the student’s various averages must be multiplied by their percentages of the total grade. The percentages will also only total 80%, since the final exam, which counts for 20% of the total grade, has not yet been taken, For quizzes, this is 15% of the total grade, so the quiz average must be multiplied by 15. For assignments, this is also 15% of the total grade, so the assignment average likewise must be multiplied by 15. For exams, this is 25% of the total grade, so the exam average must be multiplied by 25. For projects, this is 25% of the total grade, so the project average must be multiplied by 25. After all of the averages have been multiplied, they will then be summed together and divided by 80. After the current grade has been computed, **double weightedAverage(double, double, double, double)** will return the current grade. The main body of the program will then assign this value to the **currentGrade** variable.

The final function that must be defined and used is **void outputAverages(double, double, double, double, double)**. The parameter of this function accepts 5 double values, which in this case, will be the **currentGrade** variable computed by the **double weightedAverage(double, double, double, double)** function, as well as the four averages previously computed. **Void outputAverages(double, double, double, double, double)** will the output these 5 values, as well as printing out a statement for each that describes what each of them are.

Algorithm Flowchart*Diagram

Description automatically generated*

Diagram

Description automatically generated