CIS-11 Project Documentation Part 1

“Last Minute”
Leonardo Lopez & Luis Medina
05/25/2025
Test Score Calculator
Advisor: Kasey Nguyen, PhD

Part I – Application Overview

This part of the requirements document serves to present the "big picture" of the application. Here you lay out the objectives of the application, how it fits into the business process of the company, and how it relates to other software systems. The sections listed below should be included in this part of the requirements document.

Objectives

Why are we doing this?

- **What business objectives of the company will this project help achieve? Possible objectives might be reducing costs, improving the customer service, simplifying the workflow, replacing obsolete technology, piloting a new technology, and many others. Also, make sure you understand exactly how the proposed project will help accomplish the stated objective.**

The project can reduce the time and resources needed for manual work by automating the manual stuff onto a calculation-based test score, lowering the costs of manual calculation. Customers may receive quick and accurate test results from the calculator which will improve the entire experience and happiness. Increased client retention and loyalty may result from this. By simplifying the process of computing exam results and eliminating the time-consuming nature of manual work, the automatic calculator can free up time.

- **Why are we doing this project now? What will happen if we do it later?What if we do not do it at all?**

The manual method we currently use to calculate tests takes lots of time, is prone to errors, and uses a lot of resources just to do that. This procedure can be automated to cut down on labor and lower the possibility of mistakes. We must remain competitive by offering an innovative and effective method of accessing exam results since more and more students need online services. Delaying the project would mean falling behind our competitors and in general just losing business.

- **Who will benefit from this project? Do the people who will benefit from it consider it the most important improvement that can possibly be made at this time? Should we be doing a different project instead?**

Students/clients will benefit from faster and more accurate test scores. It is a worthwhile improvement as it establishes a drastic increase in efficiency and accuracy for student test results, which is a very important aspect in most students' lives and will help them immensely in viewing their test performances. I wouldn't say it's a major issue to our society but it does make it time efficient for them.

Business Process

In the future, this program can become a more sophisticated tool for calculating and reporting scores based on weighted grades, different categories, and percentage contributions to the final grade. In example quizzes can be worth 15% while exams are 85% to provide a more accurate reflection of student performance, we can adjust the programs user interface, variable, count, subroutines therefore improving grading accuracy, and changing the user's number of input/scores to be their choice. This allows for an even wider range of flexibility and tests overall performance at the fingertips for students from across the world to access. This application will serve as a pivotal tool for many students to stay on top of their academic success and be able to track and calculate their test performances on a different level.

Terminology

Score Input – Numerical Value Ranging from "0-100"

Letter Grade – Letters (A-F) assigned based on numerical value

Score List – List of all scores entered for further analysis

Total Score – Sum of all scores which is used to calculate the average

Min Score – Lowest score of all 5 entered

Max Score – Highest score of all 5 entered

Average Score – Sum of all scores on the list divided by the # of scores (5)

Subroutine – ConvertScoreToGrade(score) that performs specific task as said by function name

Branching – The use of IF-ELSE logical statements in condition of exam score to grade convertion.

Iteration – A loop that repeats the task of grading several exam scores

User Responsibilities

Provide Valid Input – Requires a value from "0-100" for all 5 exams

Follow Instructions – Only input values when program is requesting

Interpret Output – Understand each values designated grade

Statement of Functionality

Input function

- Prompt user to input five test scores
- The program will output a message asking the user to enter five test scores via the keyboard

Minimum score calculation

- Determine the minimum score of the five test scores
- The program will compare scores and find the smallest value. Store

Maximum score calculation

- Determine the maximum score of the five test scores
- The program will compare scores and find the largest value. Store

Average score calculation

- Compute the average score
- The program will add all entered test scores and then divide by five (the total number of scores)

Determination of letter grade

- Assign letter grades to each test score based on the predetermined ranges
- Conditional branching

Output function

- Display results to the console
- The program will display the above information to the console (minimum score, maximum score, average score, letter grade corresponding to the test scores

Flow chart or pseudo-code.

Include branching, iteration, subroutines/functions in flow chart or pseudocode.

CONDITIONALS TASKS, BRANCHING, ITERATIVE TASKS, SUBROUTINE

```
START
PRINT "Enter 5 scores:"
INITIALIZE
score_list as empty list SET
total_score to 0
FOR i FROM 1 TO 5 DO
PROMPT user for a score
READ score
APPEND score to score_list
ADD score to total_score
IF score >= 90 THEN
   PRINT 'A'
ELSE IF score >= 80 THEN
   PRINT 'B'
ELSE IF score >= 70 THEN
   PRINT 'C'
ELSE IF score >= 60 THEN
   PRINT 'D'
ELSE
   PRINT 'F'
END IF

END FOR
SET max_score to first element in score_list
SET min_score to first element in score_list
FOR each score IN score_list DO
IF score > max_score THEN
SET max_score to score
END IF
```

```
IF score < min_score THEN

SET min_score to score

END IF

END FOR

SET average_score to total_score divided by 5

PRINT "MAX: ", max_score

PRINT "MIN: ", min_score

PRINT "AVG: ", average_score

END
```

```
SUBROUTINE ConvertScoreToGrade(score)

IF score >= 90 THEN
   PRINT 'A'
ELSE IF score >= 80 THEN
   PRINT 'B'
ELSE IF score >= 70 THEN
   PRINT 'C'
ELSE IF score >= 60 THEN
   PRINT 'D'
ELSE
   PRINT 'F'
END IF
END SUBROUTINE
```