# **CIS11 Course Project Part 1: Documenting the Project**

Fill in the following areas (purple).

**Introduction**

**1.1** **Purpose**

Create an LC-3 program that displays the minimum, maximum and average grade of 5 test scores and display the letter grade associated with the test scores.

**1.2** **Intended Audience and Users**

The primary audience and/or users would be teachers doing their grading.

**1.3** **Product Scope**

What is the intention of this program?

The test score program should provide a fast and efficient method for teachers wanting to input and extrapolate data from their students tests.

**1.4** **Reference**

**Source Documents for the Program Requirements and Specification**

Reference Project requirements and LC-3 specifications.

**LC-3 Simulator and LC-3 Assembler**

1. https://highered.mheducation.com/sites/0072467509/student\_view0/lc-3\_simulator.html

**Companion Application Requirements Documents (If applicable)**

What other documents should be reviewed with this document?

**StudentTestGrades.doc**

**2. Overall Description**

**2.1 Product Perspective**

Primary program objectives

This program provides an easy way for teachers to drop the lowest test scores if they so choose, as well as find an average of their student’s test scores for grading purposes.

**2.2** **Product Functions**

**The overall description of functionality:**

Highlight the program functionality: Identify tasks and subtasks of the program in summary.

1. Store five test scores within their own registers. Have another register to calculate averages with.
2. Once the test scores are inputted, compare them against the highest grade threshold for each grade in order of lowest grade to highest, and if it is lower, then output the appropriate grade (or, alternatively, compare against the lowest grade threshold for each grade in order of highest grade to lowest, and if it’s higher, then output the appropriate grade).
3. Sort the grades from lowest to highest and then output the highest and lowest to the console.
4. Calculate the average of the test scores.
5. Output it to the console.
6. Through the process in step 2, determine the letter grade the average would receive and output it to the console.

**Technical functionality**

A configurable toolkit of functions including:

What are the technical functions of the program? Subroutines and operations.

Ability to sort values from highest to lowest

Ability to compare values to a range of numbers

Ability to calculate averages

2.3 **User Classes and Characteristics**

**Who are involved in this development process? Include business and technical personnel and their tasks.**

**Faculty**

To see how a class is performing and adjust curriculum accordingly.

If grading on a curve is desired, the program will allow one to easily find the class average.

**2.4** **Operating Environment**

What type of system will the application be operated on? Operating system? System types? Development platform?

The system will operate using LC-3 code and an LC-3 simulator. It will be available for use on any OS that can handle LC-3 simulators and code.

**2.5** **Design and Implementation Constraints**

Note any constraints or limitation to the application.

No higher than 5 grades can be input at a time. The system may not be able to take values greater than 100 or, somehow, lower than 0, depending on the implementation method.

**2.6** **Assumptions and Dependencies**

Note any dependencies

It is assumed that the user is familiar with handling the keyboard and mouse. It is also assumed that the required test scores will be available, since those are needed as input for the program to work properly.

***3*. External Interface Requirements**

3.1 **User Interfaces**

How will the user interface with your program? Menus? Access prompt? Links? Icons?

The user will input the necessary information through the console when prompted.

3.2 **Hardware Interfaces**

Specify hardware interface – computer types? Terminal types?

Any computer or terminal type should be adequate for the test score data entry.

3.3 **Software Interfaces**

Specify additional software interface – if any. What type of software will the application require to run?

Any software capable of running an LC-3 simulator.

3.4 Communications Interface

Does your application require web, Internet or network connectivity? If so, which browser? What type of network connection?

The application will not require any network connectivity, only the data of the five test scores.

**4. Detailed Description of Functional requirements**

**4.1 Type of Requirement (summarize from Section 2.2)**

**What are the functions? Their purposes? Inputs? Outputs? Data? Where is the data stored (internal or external to the application)?**

**Average Calculation**

Purpose: calculates an average of the five test scores

Inputs: Inputs are through the keyboard.

Processing: The 5 inputted test scores will be added together and be divided by five.

Outputs: The average of the five test scores will be outputted to the console on LC-3 simulator

Data: User database

**Minimum Calculation**

Purpose: calculates a minimum of the five test scores

Inputs: Inputs are through the keyboard.

Processing: The 5 inputted test scores will be compared to find the lowest test score

Outputs: The minimum of the five test scores will be outputted to the console on LC-3 simulator

Data: User database

**Maximum Calculation**

Purpose: calculates a maximum of the five test scores

Inputs: Inputs are through the keyboard.

Processing: The 5 inputted test scores will be compared to find the highest score.

Outputs: The maximum of the five test scores will be outputted to the console on LC-3 simulator

Data: User database

**Grade Letter Display**

Purpose: Display letter grade associated with corresponding score.

Inputs: Inputted scores should be through the keyboard.

Processing: When test scores are saved in the registers, a grading system(A=90 or above, B=80 or above, C= 70 or above, D = 60 or above, and F = 59 or below) will be applied to each test score.

Outputs: The letter grade will be outputted with the associated test score/

Data: User database.

**4.2 Performance requirements**

**What is the expected performance level of the program?**

**4.2.1** The application should be portable and possible to users of LC-3 simulator.

4.2.2 Since the application will be displaying data for only 5 test scores, the response time for a particular analysis should be not be greater than 1-2 seconds for the simulator to generate the data for the scores.

4.2.3 The database should be scalable; it must have the capacity to hold a greater number of tests in future.

**4.2.4** Error handling should be implemented and the application should be able to handle all runtime errors.

**4.3 Flow Chart and Pseudocode.**

Calculate average based off Running score and number of student tests being graded

Display AverageScore, HighScore, and LowScore to user with letter grade

Start

Variable initialization based on user input

Calculate Running, hi, low scores

While Student number <= 5

If score < low; low = score;

If score > high; low = score;

RunningScore =+ score;

Index +=1

HighScore = 0

LowScore = 0

AverageScore = 0

RunningScore = 0

Students 1 - 5 are inputted to an array

//Variable initialization

int HighScore = 0;

int LowScore = 100;

int AverageScore = 0;

int RunningScore = 0;

int StudentScores[5];

StudentScores[i] = Prompt user for StudentScores and input into array;

//Calculate low, high, and average score

for(int i; i <= StudentScores.arraysize();i++)

{

if(StudentScores[i] < LowScore)

{LowScore = StudentScores[i]}

if(StudentScores[i] > HighScore)

{HighScore = StudentScores[i]}

RunninScore =+ StudentScores[i];

}

//Calculate average

AverageScore = RunningScore/StudentScores.arraysize();// Or hardcode to 5

//Display processed scores

Display the AverageScore, LowScore, and HighScore with appropriate description and letter grades