# CIS11 Course Project Part 1: Documenting the Project

Fill in the following areas (purple).

**Introduction**

* 1. **Purpose**

The purpose of this LC-3 program is to calculate the minimum, maximum, and mean of five test scores, then display a letter grade for the resulting values.

* 1. **Intended Audience and Users**

Anyone who is interested in averaging numbers should be interested in using this program. The primary audience for this program is intended for anyone attending schools with a letter grade system, but the ability to calculate the average and ranges of five numbers can be applied to any users interested in such a function.

* 1. **Product Scope**

What is the intention of this program?   
The intention of this program is to help students quickly calculate the averages of their scores on their own instead of having to wait from an official tally from their teachers. This program should be simple enough for anyone to follow without prior experience to an arithmetic calculator.

* 1. **Reference**

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

Reference Project requirements and LC-3 specifications.

1. LC-3 Tutorial services  
   http://www.lc3help.com/
2. LC-3 editor and simulator  
   <https://highered.mheducation.com/sites/0072467509/student_view0/lc-3_simulator.html>
3. Draw.io flowchart creator  
   https://www.draw.io/

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

What other documents should be reviewed with this document?

1)     Calculator.drawio

2)     CIS11 Course Project Documentation.doc

**2. Overall Description**

**2.1 Product Perspective**

Primary program objectives  
Test score calculator uses addition and division subroutines to calculate an average and min-max values based on user input of five numbers to the hundredth place.

* 1. **Product Functions**

**The overall description of functionality:**

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

Test Score Calculator can output to console:

Maximum score

Minimum score

Average Score

Letter grade equivalence of average score. (0-50 = F, 60-69 = D, 70-79 = C, 80-89 = B, 90-100 = A)

**Technical functionality**

A configurable toolkit of functions including:

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

Add scores together and divide by 5 to output an average

Detect the highest value number from a range of 5 scores using addition and subtraction

Detect the lowest value number from a range of 5 scores using addition and subtraction

Convert an average value into a letter grade using subtraction to convert hexadecimal value into ASCII

* 1. **User Classes and Characteristics**

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

**Main Programmers**

Outline and code the actual code

**Supervisor**Reviews code, suggests improvements and documents as necessary (Psuedocode, reports, comments, etc).

* 1. **Operating Environment**

What type of system will the application be operated on? Operating system? System types? Development platform?   
This program is intended for use in any computer that meets the minimum requirements to run LC-3 simulator, with general use to the public, preferably on a Windows OS.

* 1. **Design and Implementation Constraints**

Note any constraints or limitation to the application.

Users may only input up to five test scores. This program will accept only scores up to three digits. This program cannot accept negative numbers. In the future, subroutines can be included to check for inputs for values that are not numbers from 0 to 9. A warning string can then be displayed to inform the user of their error and to retry their input accordingly.

* 1. **Assumptions and Dependencies**

Note any dependencies

This program accepts inputs up to the number 125. The program accepts ASCII input.

***3*. External Interface Requirements**

* 1. **User Interfaces**

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

The user will input their grades, one at a time, in the LC-3 simulator console. The user must press enter after inputting a grade before inputting the next. No graphics are used, only ASCII characters and numbers.

* 1. **Hardware Interfaces**

Specify hardware interface – computer types? Terminal types?   
Any computer that supports Windows operating system and can run LC-3 simulator can run this application.

* 1. **Software Interfaces**

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

LC-3 Simulator.

* 1. **Communications Interface**

Does your application require web, Internet or network connectivity? If so, which browser? What type of network connection?  
  
This program does not require an internet connection.

**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)?**

Score input:

Purpose: Accepts user input of test scores to be calculated.

Inputs: Inputs are through the keyboard and mouse clicks.

Processing:  The input is processed with addition and division subroutines to calculate the desired results. Each inputted score is stored and compared to the current MIN or MAX to be replaced if needed. Once all five scores are inputted, all scored are added up and then divided by five to calculate the average. Subroutines are also in place to store values by the tens or hundreds as needed.

Outputs: The average, minimum, and maximum score will be displayed on the console, along with a letter grade of the resulting average.

Data: User input from the keyboard.

**4.2 Performance requirements  
 What is the expected performance level of the program?**

**4.2.1** This is a lightweight program that should run on any device capable of running Windows and LC-3 simulator.

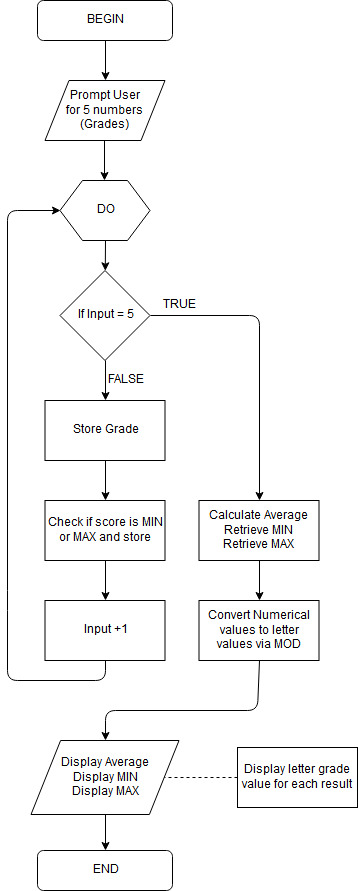
4.2.2 This application will output results immediately once the required inputs are collected.

4.2.3 In order to input a new set of scores, the program must be reset.

**4.2.4** Errors can occur if the user inputs a negative number or letters. Restarting the program will fix these errors and give the user another chance to input proper scores.

**4.3 Flow Chart and Pseudocode.**

|  |
| --- |
|  |
|  |  |

Grade Calculator pseudocode:

1. Prompt user for 5 test scores with a PUTS string.
2. GETC to collect input from user.
   1. Store each number as its own digit for ones, tens, and hundreds place then put them together as a total value
3. Check each input as a MIN or MAX
   1. For MIN, check with BRzp. Skip if nz. Store result
   2. For MAX, check with BRnz. Store result.
4. Loop 5 times with a counter until all inputs are collected.
5. Average inputs with a Division subroutine. Store result.
6. Retrieve the stored results of MIN, MAX, and AVERAGE.
   1. Use BRz to check if the last digit is zero while looping through the results to check the range of the letter grade.
   2. Check the tens (or hundreds digit if 100) to determine what grade letter should be assigned.
7. Use an ASCII conversion subroutine to convert the results into a grade letter average by subtracting 48.
8. PUTS the results out onto the console, along with String prompts explaining what each result is, either MIN, MAX, or AVERAGE.