# CIS11 Course Project Part 1: Documenting the Project

Fill in the following areas (purple).

**Introduction**

* 1. **Purpose**

This program takes 5 inputs (test scores) in from the user then calculates and outputs the average, minimum and maximum.

* 1. **Intended Audience and Users**

The primary audience/users are people who want to collect data for grades earned in a class, like teachers, professors, etc.

* 1. **Product Scope**

What is the intention of this program?

The test score calculator is intended to provide teachers/professors with a useful tool that allows them easily find out the average, minimum and maximum scores for a certain exam. It is expected to provide for accurate results than calculating the scores by hand.

* 1. **Reference**

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

Reference Project requirements and LC-3 specifications.

1. LC-3 Editor
2. LC-3 Console
3. LC-3 Simulator  
   <http://www.lc3help.com/downloads.htm>
4. Test scores to input into the program

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

What other documents should be reviewed with this document?

LC-3 Editor

LC-3 Console

LC-3 Simulator  
<http://www.lc3help.com/downloads.htm>

**2. Overall Description**

**2.1 Product Perspective**

Primary program objectives

The Test Score Calculator provides:  
An easy to use interface  
A reliable and accurate calculator  
Quick output of the results

* 1. **Product Functions**

**The overall description of functionality:**

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

Overflow control for larger number values

compact and efficient design

* 1. **User Classes and Characteristics**

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

**Samuel**

Begin project documentation

Code the multi character input subroutine

Code the average subroutine

**Randy**

Code maximum function and number to ascii output function

**Jason**

Assist with project documentation

Code the minimum subroutine

Assist with multi character input subroutine

* 1. **Operating Environment**

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

The application will be operated on windows

The application will be executed through the LC-3 editor, simulator, and console

* 1. **Design and Implementation Constraints**

Note any constraints or limitation to the application.

The LC-3 suite of applications are required to execute the program.

* 1. **Assumptions and Dependencies**

Note any dependencies

The user must know how to open the program in the LC-3 editor, how to assemble the project, how to load and run the project in simulator and how to input data into the console. Basic knowledge of windows is also assumed.

***3*. External Interface Requirements**

* 1. **User Interfaces**

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

The user will run the assembled program through the simulator. They will then open the console where they will be prompted to enter 5 test scores. After the 5 test scores have been entered, the console will output the minimum, maximum, and average test scores.

* 1. **Hardware Interfaces**

Specify hardware interface – computer types? Terminal types?   
computer

* 1. **Software Interfaces**

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

The software will only run on a windows machine. And the LC-3 applications will be required to open, run, and use the software

* 1. Communications Interface

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

 No internet connectivity is required to run this application.

**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)?**

**Test scores requirement**

Purpose: Obtain test scores from user input

Inputs: Inputs are through the keyboard

Processing: The input is stored and calculated

Outputs: After the minimum, maximum, and average have been calculated the results are output to the console

Data: User database

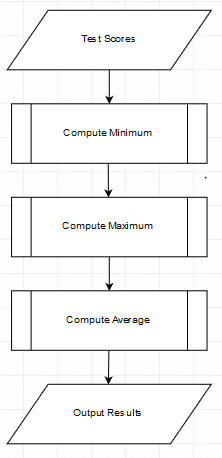
**4.2 Performance requirements  
 What is the expected performance level of the program?**

4.2.1 The application should be user friendly and easy to learn so that it’s accessible to as many people as possible

4.2.2 The expected time to output from the time the user data is inputed should be no more than 2 seconds

4.2.3 The application should be error check and verify whether the user input is within bounds

**4.3 Flow Chart and Pseudocode.**



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

Pseudocode

# Num to ASCII

To print out the results, the input will be a number from the average, min and max function.

First, divide by 100. Retrieve the quotient. This will be the hundredth’s place.

Next, divide by 10. Retrieve the quotient. This will be the ten’s place. Take the remainder, this will be the one’s place.

Store the results in registers. Put each one in R0 and output the character.

# Min

Load test values into registers

Subtract score 2 from score 1, check if negative or positive

If negative, score 1 is the target, move to compare 1 and 3

If positive, score 2 is the target, move to compare 2 and 3

Continue this pattern until the correct branch is found to determine the smallest value

Branch to one of the 5 labels relevant to the score register, store the register value in MIN label

# Max

Retrieve starting address of test score array

Immediately store the first value of the array as the **‘MAX’.** Negate the max value

Use an increment counter to loop through the rest of the elements in the array, adding the value in the array with the MAX value. If it is negative, then we know that the max was bigger in that case. However, if it was positive, then we make the swap to put in the new value as the max. Loop until end of array. Store max value in MAX label.

# Average

Load 5 test scores, call the TOTAL subroutine to add all of them together then store value in T

Then divide the the total by 5 to get the average

# Multi Character Input

We’re going to use a pointer to the array to allow an input of 3 characters. Each char entered is going to be stored in the array and the array index will be increased by 1 until it reaches 3

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