# CIS11 Course Project Part 1: Documenting the Project

**Introduction**

* 1. **Purpose**

This document details the functionality and requirements.

* An LC - 3 program which displays: minimum, maximum, and average grade of five test scores and displays the letter grade associated with those test scores
* User will input test scores
* User will receive an output of a score sheet onto their console
* Criteria to fulfill:

1. Contain appropriate addresses: origination, fill, array, input and output. (20 points)
2. Display minimum, max, average values/grades in console. (20 points)
3. Use appropriate labels and comments. (20 points)
4. Contain appropriate instructions for arithmetic, data movement and conditional operations. (40 points)
5. Comprise of 2 or more subroutines and implement subroutine calls. (20 points)
6. Use branching for control: conditional and iterative. (30 points)
7. Manage overflow and storage allocation. (20 points)
8. Manage stack: include PUSH-POP operation on stack. (20 points)
9. Include save-restore operations. (30 points)
10. Include pointer (20 points)
11. Implement ASCII conversion operations (30 points)
12. Use appropriate system call directives. (10 point)
13. Testing (20 points): Test the program using the below values
    1. **Intended Audience and Users**

Program is intended for individuals who would like a tool to faster calculate average, low and high numbers. Program will only be able to work with 5 numbers or test scores.

* 1. **Product Scope**

The intention of this program is to make it easier for the user to analyze and organize the results of tests via displaying the maximum, minimum, and mean of multiple test scores and automatically convert said scores into a letter grade equivalent.

* 1. **Reference**

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

Reference Project requirements and LC-3 specifications.

**The program contains the following criteria:**

**1. Contain appropriate addresses: origination, fill, array, input and output.**

**2. Display minimum, max, average values/grades in console.**

**3. Use appropriate labels and comments.**

**4. Contain appropriate instructions for arithmetic, data movement and conditional operations.**

**5. Comprise of 2 or more subroutines and implement subroutine calls.**

**6. Use branching for control: conditional and iterative.**

**7. Manage overflow and storage allocation.**

**8. Manage stack: include PUSH-POP operation on stack.**

**9. Include save-restore operations.**

**10. Include pointer.**

**11. Implement ASCII conversion operations**

**12. Use appropriate system call directives.**

**13. Testing program to ensure it runs successfully.**

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

What other documents should be reviewed with this document?

1. CIS11 Course Project Part 1 FINAL-1.docx
2. CIS11 Progect Part 1 Team Task Assignment-1.docz

**2. Overall Description**

**2.1 Product Perspective**

Primary program objectives  
  
This program provides:  
  The ability to display the maximum, minimum, and average grade of 5 test scores that are input by user  
  Displays the average test score and letter grade equivalent

* 1. **Product Functions**

**The overall description of functionality:**

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

The program will require that the following tasks be completed:

* Allows up to 5 user input values within the accepted range.
* Compares all input values to derermine the lowest value.
* Compares all input values to determine the highest value.
* Displays organized results on console

**Technical functionality**

A configurable toolkit of functions including:

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

* Ability to retrieve and temporarily store input values.
* Applies arithmetic instruction to come up with result.
* Compares and contrasts values until desired value (average, minimum, or maximum value) is obtained.
  1. **User Classes and Characteristics**

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

* **All people in this project are in the development process: Kevin, Brian, Jasmine, Julissa**
* **Project guidelines given by Moreno Valley College**
* **Kevin: Manages the programming throughout the LC 3 simulator and Pseudo Code**
* **Brian: Manages the flowchart in how the program should be ran**
* **Jasmine: Wrote documentation and program editing to help bug testing**
* **Julissa: Evaluation of the code to see if it could be ran smoother along with finishing documentation**

**Faculty**

Plan and teach curricula  
Responsible for creating idea of program **Students**

Planning and documenting project  
Responsible for creating the program through LC-3  
Representing project file through GitHub  
Participating in Team Evaluation

* 1. **Operating Environment**
* **The application will be operated on a Windows 7 and higher operating system.**
* **More than likely a Windows 10 Computer**
* **The application is being developed on a Windows 10 computer, created in an LC - 3 compiler for our assembly code**
  1. **Design and Implementation Constraints**
* **The user may only input five scores at one time - no more, no less.**
* **The average won’t be a true estimate since it won’t include decimals nor would it round correctly as it would technically just round down thus not representing a true 100% score.**
* **The user interface would be simple yet could always be more user friendly**
* **User input error**
  1. **Assumptions and Dependencies**

Note any dependencies

It is assumed that the values to be entered will only fall between 0-100. User is to understand the letter grading scale:

* A : 90 - 100
* B: 80 - 89
* C: 70 - 79
* D: 60 - 69
* F: 0 - 59

***3*. External Interface Requirements**

* 1. **User Interfaces**
* **The user will be prompted with a menu which will display a text box:**
* **Textbox will ask for the scores**
* **The user will then input the test scores**
* **The program will then display the scores as a letter grade, along with the: min, max, and average of the five scores**
  1. **Hardware Interfaces**
* Must include keyboard and screen display to function
* Must be hardware that runs LC-3 simulator

Specify hardware interface – computer types? Terminal types?

* 1. **Software Interfaces**

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

* **No additional software will be required**
* **Only LC - 3 simulator will be required**
  1. **Communications Interface**

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

* Program will not be required to connect to the web or internet

**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)?**

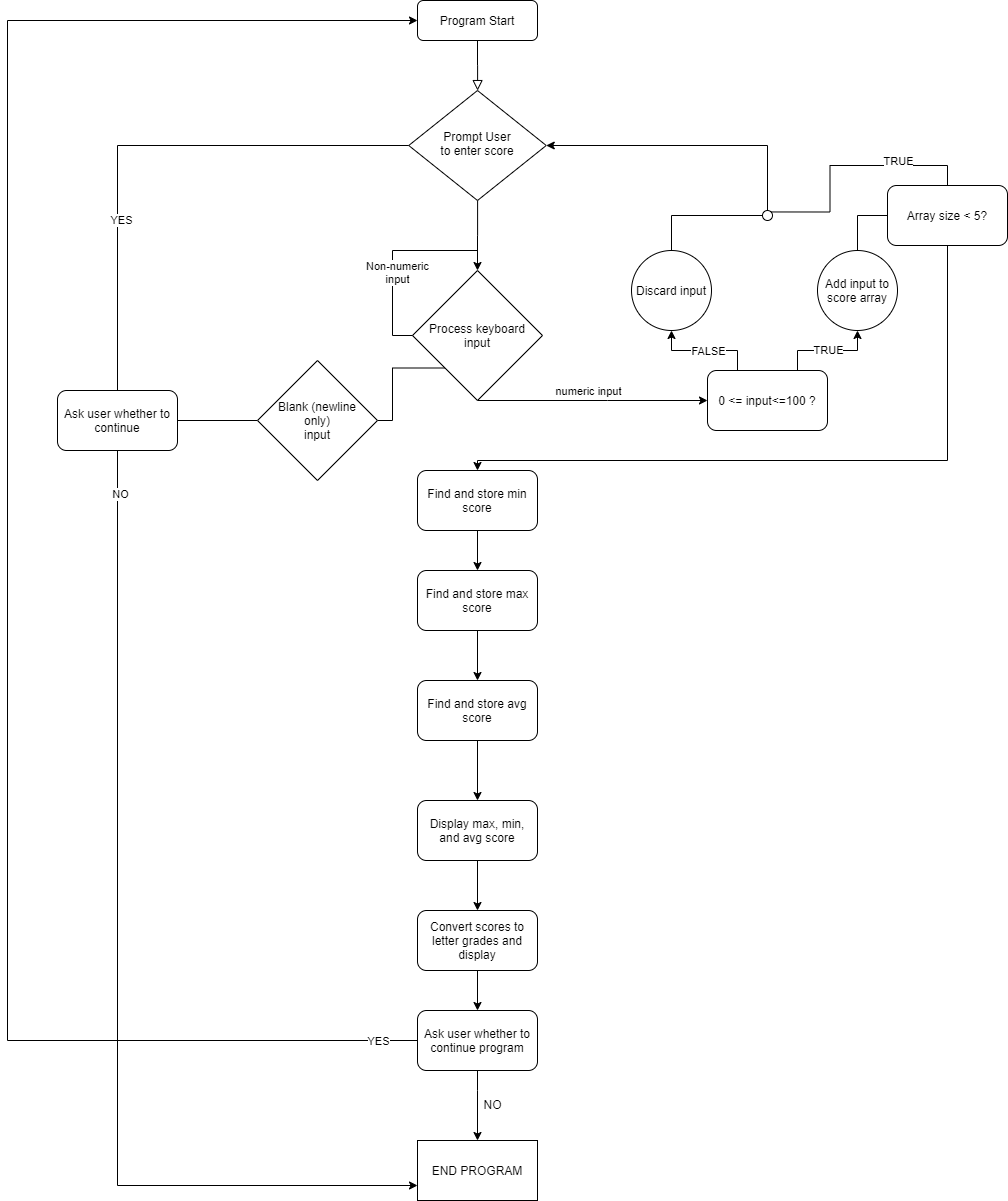
**Test Score Calculator Requirement**  
 Purpose: Provides the minimum, maximum, and average grade results  
 of 5 test scores

Inputs: Inputs are through the keyboard and mouse clicks.  
  
 Processing: The input is verified if the test score meets the in-range   
 number requirements  
  
 Outputs: The correct inputs will display the minimum, maximum, and   
 average score and letter grade equivalence (0-50=F, 60-69=D, 70-79=C,   
 80-89=B, 90-100=A). If the output is incorrect, there will be an error   
 message displayed.

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

* **The performance of the application should be moderately fast to fast in terms of giving the response back to the user.**
* **The application will be displaying: min, max, and average of the five scores**
* **The response time for calculations and storing the values the user has input along with doing the calculations should take the majority of the time yet it should not take greater than 8 seconds.**
* **The application should be able to handle most if not all runtime errors unless the user has input a ludicrous amount or had an error on their part.**

**4.3 Flow Chart and Pseudocode.**

****

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

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

Pseudocode:

PROGRAM START

WHILE TEST SCORE ARRAY SIZE < 5

CLEAR TOTAL

PROMPT USER TO ENTER TEST SCORE (or submit a blank line to END the program immediately)

WHILE NEXT CHARACTER IS NOT NEWLINE

GET CHARACTER FROM KEYBOARD

CONVERT FROM ASCII

IF CHARACTER IS 0, 1, 2, 3, 4, 5, 6, 7, 8, OR 9

PUSH INTEGER VALUE ONTO INPUT STACK

CONVERT BACK TO ASCII

DISPLAY CHARACTER BACK TO CONSOLE

CONTINUE LOOPING

IF INPUT STACK IS EMPTY

PROMPT USER “Would you like to end the program?”

IF NO

PROMPT USER “Clear all previously entered scores?”

IF YES

CLEAR ARRAY, REGISTERS, AND RETURN TO PROGRAM START

ELSE CONTINUE LOOP

ELSE END PROGRAM

WHILE INPUT STACK IS NOT EMPTY

COPY TOP OF STACK TO REGISTER

ADD REGISTER TO TOTAL

POP INPUT STACK

IF TOTAL < 0 OR TOTAL > 100

INFORM USER THAT INPUT IS INVALID

CLEAR INPUT STACK

CONTINUE

ADD TOTAL AS NEW ARRAY ELEMENT

LOAD ADDRESS OF ARRAY TO REGISTER

SET MINIMUM VALUE TO FIRST ELEMENT

SET MAXIMUM VALUE TO FIRST ELEMENT

LOOP 4 TIMES:

ADVANCE TO NEXT ARRAY ELEMENT

IF CURRENT ELEMENT > MAXIMUM VALUE

SET MAXIMUM VALUE TO CURRENT ELEMENT

CONTINUE

IF CURRENT ELEMENT < MINIMUM VALUE

SET MINIMUM VALUE TO CURRENT ELEMENT

CONTINUE

STORE MAXIMUM AND MINIMUM VALUES

POINT BACK TO FIRST ELEMENT OF ARRAY

ADD FIRST ELEMENT TO TOTAL

LOOP 4 TIMES:

ADVANCE TO NEXT ELEMENT

ADD CURRENT ELEMENT TO TOTAL

CONTINUE

DIVIDE TOTAL BY 5 AND STORE AVERAGE

DISPLAY “Minimum score: “

LOAD MINIMUM

TOASCII(MINIMUM)

DISPLAY “Maximum score: “

LOAD MAXIMUM

TOASCII(MAXIMUM)

DISPLAY “Average score: “

LOAD AVERAGE

TOASCII(INPUT)

LOAD ADDRESS OF ARRAY

LOOP 5 TIMES:

TOASCII(CURRENT ELEMENT)

IF CURRENT ELEMENT < 60

PRINT “F”

ELSE IF CURRENT ELEMENT < 70

PRINT “D”

ELSE IF CURRENT ELEMENT < 80

PRINT “C”

ELSE IF CURRENT ELEMENT < 90

PRINT “B”

ELSE PRINT “A”

PROMPT USER ON WHETHER TO ENTER MORE SCORES

IF YES

CLEAR REGISTERS, STACK, AND ARRAY

GO BACK TO PROGRAM START

ELSE END PROGRAM

FUNCTION TOASCII(INPUT)

IF INPUT = 100

PRINT INPUT TO CONSOLE

ELSE

IF INPUT > 10

DIVIDE INPUT BY 10

CONVERT QUOTIENT TO ASCII

PRINT VALUE TO CONSOLE

DIVIDE INPUT BY 100

CONVERT QUOTIENT TO ASCII

PRINT VALUE TO CONSOLE