# CIS11 Course Project Part 1: Documenting the Project

**Introduction**

* 1. **Purpose**
* This LC-3 program displays the minimum, maximum and average grade of 5 test scores and display the letter grade associated with the test scores.
  1. **Intended Audience and Users**
* Students, Teachers, and Instructors
* Applicable to anyone looking to utilize test scores for an intended purpose
  1. **Product Scope**
* The intention of this program is to allow teachers, students, and instructors to calculate the minimum, maximum, and averages of their test scores.
* By utilizing the minimum, maximum, and averages of test scores, users of this program can adapt or implement this program for more effective results.
* For example, students can utilize this program to see which areas need improvement when receiving back test scores. Teachers can utilize the program to get a better understanding of how their class is performing and can also see whether a specific individual is struggling or has a better understanding of a material than the rest of the class. Business can utilize this program to baseline their employee understanding of their knowledge if they hold reviews for knowledge on safety or equipment usage.
  1. **Reference**

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

1. [https://github.com](https://github.com/)
2. <https://highered.mheducation.com/sites/0072467509/student_view0/lc-3_simulator.html>
3. [LC 3 Simulator Windows Version 3.01 (385.0K)](https://highered.mheducation.com/sites/dl/free/0072467509/104652/LC301.exe)
4. [LC-3 Unix Simulator (56.0K)](https://highered.mheducation.com/sites/dl/free/0072467509/104652/lc3tools_v12.zip)
5. LC3Edit.exe
6. Simulate.exe

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

1. CIS11 Project Part 1 Team Task Assignment.docx
2. Pseudocode.docx
3. Flowchart.docx
4. CIS11 Course Project Part 2 LC3 Programming.docx
5. CIS11 Final Project.asm
6. CIS11 Final Project.objZ

**2. Overall Description**

**2.1 Product Perspective**

The primary objectives are:

* to make a program that is efficient
* clearly state what the program is asking of the user
* simple to use for both teachers and students
* calculate the scores accurately
* display the results after calculations
  1. **Product Functions**
* The user will be prompted to key in numbers to represent the test scores
* The program will save user input as an array
* There will be input validation to make sure the user keys in numbers between 0 - 100
* The program will calculate the maximum, minimum, and average score
* It will then display the maximum, minimum, average score and letter grade equivalence (0 – 50 = F, 60 – 69 = D, 70 – 79 = C, 80 – 89 = B, 90 – 100 = A) on the console.

**Technical functionality**

A configurable toolkit of functions including:

* The program will contain addresses: origination, fill, array, input and output.
* Display minimum, max, average values/grades in console
* Comments to aid programmers in better understanding the code
* Contain appropriate instructions for arithmetic, data movement and conditional operations.
* The program will be comprised of subroutines and implement subroutine calls.
* The use of branching for control: conditional and iterative (e.g. asking for multiple test scores and input validation)
* Manage overflow and storage allocation.
* Manage stack: include PUSH-POP operation on stack.
* Includes save-restore operations.
* Includes pointer
* Implements ASCII conversion operations
* The use of appropriate system call directives.
  1. **User Classes and Characteristics**

**Developers**

Guillermo Castillo

Josue Salcedo

Omar Lucatero

**Educators**

* Teachers will be using this program to calculate a student’s grades
* They can help in providing feedback after testing the program to make improvements if necessary

**Students**

* Students will be using this program to calculate their grades or “what if” grades to estimate a possible average
* They can help in providing feedback after testing the program to make improvements if necessary
  1. **Operating Environment**

The application will run on the following operating systems:

* Windows 7
* Windows 8
* Windows 10

The simulator used is the:

* LC 3 Simulator Windows Version 3.01 (385.0K)
  1. **Design and Implementation Constraints**
* Access to a computer with windows or linux
* Computer must be capable of running the LC-3 Assembler and console
* Will only calculate 5 test scores
  1. **Assumptions and Dependencies**
* It is assumed that the user is familiar with an internet browser and also familiar with handling the keyboard and mouse.
* Since the LC-3 application does not automatically come with Windows or Linux, there is a need of internet access to download the application to run the program
* It is assumed that the user of the program will be able to save files from their current program as the appropriate file types.
* The user is expected to know how to run both the assembler and simulator including loading the files.

***3*. External Interface Requirements**

* 1. **User Interfaces**

The user must download an LC-3 simulator for their compatible software from the website provided in references. Once downloaded, the user will be able to access and download the file and run it on the simulator, not the editor. The file will be provided in a separate folder.

* 1. **Hardware Interfaces**
* Student Lab computers
* Student personal computers
* Educator’s classroom computer
  1. **Software Interfaces**

The application will run on the following operating systems:

* Windows 7
* Windows 8
* Windows 10

The simulator used is the:

* LC 3 Simulator Windows Version 3.01 (385.0K)
  1. Communications Interface

 The application only requires the web to download the LC-3 simulator. Any browser can make it possible to search for the download.

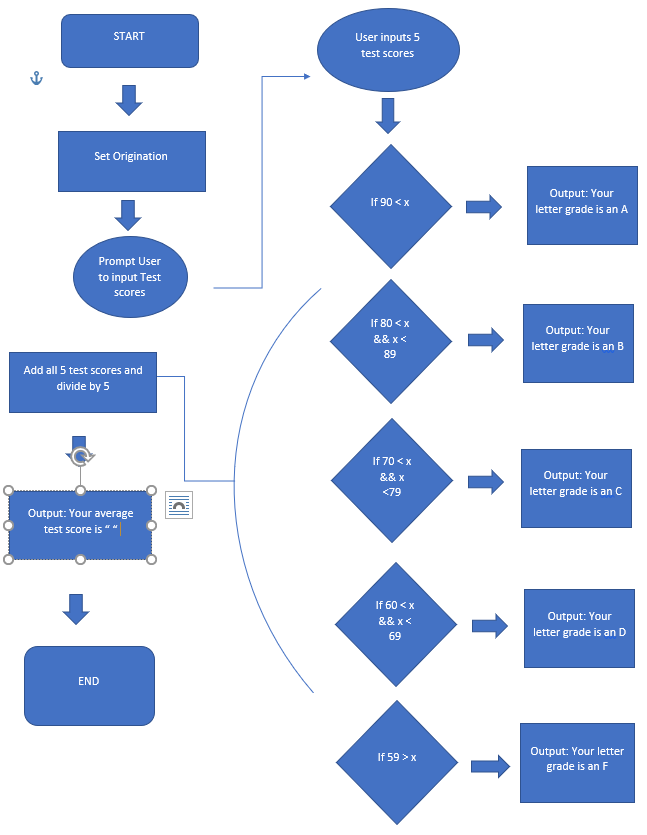
**4. Detailed Description of Functional requirements**

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

* This program creates 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.
* It is meant to make a few calculations fast and simple for students and teachers
* Inputs: Inputs are through the keyboard and mouse clicks.
* Outputs: Will display the calculated results onto the screen through the console after the user has entered the scores accordingly
* Data will be stored internally through the use of registers but the files will be loaded from external sources
* Processing:  The input is verified by checking if the member already exists in the database.

**4.2 Performance requirements**

* **The application should be able to run on most computers running recently supported Windows operating systems.**
* **The program should run smooth and efficient**

**4.3 Flow Chart and Pseudocode.** 

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

 Test Score Calculator:

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.

|  |  |
| --- | --- |
| **High-level Language** | **LC3** |
| #include<iostream> | .ORIG x3000 |
| Cout << “Enter your five test scores.” << endl; | PUTS |
| Cin>> 5 Scores | GETC, ; OR IN |
|  |  |
|  |  |
|  |  |
| If grade(90 < x) | LD R0, x90 |
| Cout << “A”; | LD R1, x  NOT R1, R1  ADD R1, R1, #1  ADD R0, R0, R1  BRn BGRADE |
| BGRADE  If grade(80 > x) | LD R1, x80  NOT R1, R1  ADD R1, R1, #1  ADD R0, R0, R1  BRn CGRADE |
| Cout << “B”; |  |
| CGRADE  If grade(90 > x) | LD R1, x70  NOT R1, R1  ADD R1, R1, #1  ADD R0, R0, R1  BRn DGRADE |
| DGRADE  If grade(90 > x) | LD R1, x60  NOT R1, R1  ADD R1, R1, #1  ADD R0, R0, R1  BRn FGRADE |
| Cout << “D”; | .END |

**;Step 1: Set Origination Address**

**;Step 2: START**

**;Step 3: Display prompt using PUTS**

**;Step 4: Get user input**

**;Step 6: Convert ASCII input to hex**

**;Step 7: Input validation**

**;Step 8: Branch into each sub if other greater than 100; 100-90, 89-80. Etc.**

**;Step 11: Display the letter grade.**

**;Step 12: END code**