# CIS11 Course Project Part 1: Documenting the Project

Fill in the following areas (purple).

**Introduction**

**1.1**  **Purpose**

**Summarize program goals and objectives.**

This program’s purpose is to display 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/user**

This program is intended to be used in a school setting, so the primary audience/users are students and teachers.

**1.3**  **Product Scope**

**What is the intention of this program?**

The intention of this program is to allow students/teachers to gather more information out of test scores.

**1.4**  **Reference**

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

**Reference Project requirements and LC-3 specifications.**

**Input & Output**

**LC-3 Simulator:**<https://highered.mheducation.com/sites/0072467509/student_view0/lc-3_simulator.html>

**Project Requirements:**

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.

**Input:** User is prompt to input the test scores.

**Output:** Display **maximum, minimum, average scores** **and letter grade equivalence** (0 – 50 = F, 60 – 69 = D, 70 – 79 = C, 80 – 89 = B, 90 – 100 = A) on the console.

**The program must fulfill the following criteria:**

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.

**2. Overall Description**

**2.1 Product Perspective**

**Primary program objectives and goals.**

**Data type**

* Prompt the user to input 5 integers ranging from 0 – 100.
* Calculate the minimum, maximum, average scores, and letter grade equivalence.
* Display the results of these calculations onto the console.

**2.2** **Product Functions**

**The overall description of functionality:**

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

The program will ask for 5 integers and will individually check if each integer is in the range of 0 – 100. If not in this range of 0-100, display a message indicating an error and end the program. If the inputs are valid, the program will check each input to categorize them in the ranges of 0 – 50, 60 – 69, 70 – 79, 80 – 89, and 90 – 100 to give each test score the appropriate letter grade. Once the program receives the 5 inputs, it begins the calculations of minimum, maximum, and average. The program displays the results on the console and ends the program.

**Technical functionality**

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

Subroutines

* PROMPTINPUT: A subroutine used to ask the user for 5 integers.
* LETTER: A subroutine that converts the given integers into their correlating letter grade.
* CALCAVERAGE: A subroutine that calculates the average of the scores.
* CALCMAXIMUM: A subroutine that calculates the maximum of the scores.
* CALCMINIMUM: A subroutine that calculates the minimum of the scores.
* PUSH: A subroutine that adds a piece of data on top of the stack.
* POP: A subroutine that removes a piece of data from the stack.

2.3  **User Classes and Characteristics**

**What type of users are involved in this development process? Include business and technical personnel and their tasks.**

Students: Anthony Ocampo, Angel Alcazar, Paolo Ramos:

Programming, Program documentation, and Testing.

**2.4** **Operating Environment**

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

**How should the application be used? Simulator version?**

This program was developed and operated on Windows with the use of the LC-3 simulator. Do note that the LC-3 simulator is available for UNIX, but without testing on that platform, functionality is not guaranteed.

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

**Note any constraints or limitations to the application.**

The program can only take in and operate with 5 test scores. This program also requires the test scores to be whole numbers, so the program cannot use decimals for greater precision. The program is also limited to only 3 operations ( minimum, maximum, and average) and can be expanded on to provide better data analysis.

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

**Note any dependencies.**

**Is this application dependent on other applications or services? Browser? Web services? Simulator?**

The application is dependent on the LC-3 Simulator software to run the code and display the console.

***3*. External Interface Requirements**

3.1  **User Interfaces**

**How will the user interface with your program? Menus? Console?**

After running the program in the LC-3 simulator, the user can interact with the program through the console. The user will follow the prompts and input data with a keyboard when asked. At the end, the user will be given the minimum, maximum, and average grade for the user as well as corresponding letter grades for the given input.

3.2  **Hardware Interfaces**

**Specify hardware interface – computer types? Terminal types?**

Any windows computer (laptop/desktop) with the capabilities to acquire and run the LC-3 simulator.

3.3  **Software Interfaces**

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

The program is entirely dependent on the Windows LC-3 assembler and simulator and will require that software to run.

3.4  Communications Interface

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

If the computer does not have the LC-3 Simulator installed, an internet connection would be required to download the software, but will not be required thereafter to run the program.

**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)?**

* Purpose: Ask for 5 test scores and provide the maximum, minimum, average scores and letter grade equivalence.
* Inputs: The user is prompted to input 5 integer test scores ranging from 0 – 100.
* Outputs: The user is displayed prompts for test scores, corresponding letter grades, and the results from the maximum, minimum, average calculations. If detected, will also display error messages for invalid inputs.
* Data: The data is stored internally in arrays.

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

Given the small complexity and size, the program is expected to run within seconds. The program is also expected to check for errors and end the program instead of calculating and displaying invalid data.

**A screenshot of a computer screen

Description automatically generated4.3 Flow Chart OR Pseudocode.**