# CIS11 Course Project Part 1: Documenting the Project

Fill in the following areas (purple).

**Introduction**

* 1. **Purpose**

The purpose of this code is to allow students to calculate the average of their scores while also being able to see what the highest and lowest scores are respectively. Based on calculations coded into the code, this program will then be able to output a letter grade that corresponds to the percent average found. This program requires the user to know their scores—preferably the percentage in the occurrence that the points are along the lines of 5 out of 5 (thus 100%) instead of 5 out of 100 (which would heavily skew the calculations because this would be a very low 5%).

* 1. **Intended Audience and Users**

The primary audience for this program would be current students, so they can calculate their grades as the semester goes forward. Users can also include educators and parents in the occurrence that this program is used in elementary through high school settings as well. Given that it is expected that the user will have all the scores at hand, this program still heavily leans towards independent students who have access to all their scores.

* 1. **Product Scope**

The intention of this Test Score Calculator program is to allow students to easily be able to not only see their current grade, but also find what scores they may need to get the letter grade they’re aiming for. This program would aim to act in part like the gradebook on canvas, and would more than likely be a downloadable mini program that could run with command prompt. This way students would be able to check their potential grades even if they don’t have access to the internet and/or canvas.

* 1. **Reference**

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

Reference Project requirements and LC-3 specifications.

**Example**

 1) CIS11 Course Project Criteria

https://rccd.instructure.com/courses/35450/files/6235135?wrap=1

 2) CIS11 Course Project Points Breakdown

https://rccd.instructure.com/courses/35450/assignments/871517?module\_item\_id=1924330

 3) CIS Project Participants

      https://docs.google.com/document/d/11nPrwkA1oWACglbjheiIq\_WzGnM5lgUveaFaZ0Rbihc/edit

 4) Course Project Module Timeline

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

What other documents should be reviewed with this document?

1)     Flowchart Maker and Online Diagram Software  
 https://app.diagrams.net/#LGrade%20Program%20Flowchart.html

2)     CIS11 Group Project team document  
https://docs.google.com/document/d/1\_LmQikk10IcvFb83m0RjLvdF1FjS\_RgbT4tSZhNdbBM/edit

3)     Applicable portions from previous labs

4) “A Gentle Guide to Using the LC-3 Simulator”  
https://www.seas.upenn.edu/~cit593/resources/lc3guide.html

**2. Overall Description**

**2.1 Product Perspective**

Primary program objectives

This Test Score Calculator provides:

* A definitive letter grade.
* A percentage value based on the input scores.
* The highest score from the scores provided.
* The lowest score from the scores provided.
  1. **Product Functions**

**The overall description of functionality:**

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

Tasks:

1. This program will first and foremost calculate your grade based on the scores provided.
   1. Will output the grade in letter form.
   2. Will output the grade in percentage form.

Sub-tasks:

1. Will find the lowest score of the scores provided.
2. Will find the highest score of the scores provided.
3. Will find the sum of the scores provided.
4. Will find the average of all the scores.

**Technical functionality**

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

* Ability to calculate a decimal value average.
* Ability to add and divide multiple values.
* Ability to output a letter value based on percentage calculated.
* Flexible system to allow multiple values to be input for calculations.
  1. **User Classes and Characteristics**

**Students**

Allen Trujillo

* Code Compiler
* Core Programmer

Ryan Lane

* Advisor
* Tester
* Program Editor

Alejandra Andrade

* Flowchart/Pseudocode Designer
* Technical Writer
  1. **Operating Environment**

This application would be operated on Java for command line usage. It would be compatible with any operating system and system type.

* 1. **Design and Implementation Constraints**

For this program to achieve its designated goal, it must be assumed that students both know and have access to their current grades. This may act as a constraint if students are limited in their access to their information, or if they have incorrect information they would be inputting. Another constraint would be assuring they have Java installed on their system so that they may be able to run it in the command prompt.

* 1. **Assumptions and Dependencies**As mentioned above, it is assumed that the user has access their information. However, a dependency from our end is the fact (based on testing and continued improvements) that our math is correct, and that the problem is running as effectively as possible.

***3*. External Interface Requirements**

* 1. **User Interfaces**

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

As this program is a very simply LC-3 program, the user will interface with it through Java and the command prompt. There will be no menus, access prompts, links, or icons. There will be a prompt that requests information when they open the program, but it will very a very simple program to do a very simple task.

* 1. **Hardware Interfaces**

The hardware required is a working device that can run the command prompt.

* 1. **Software Interfaces**

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

The preferred software would be a Java based command line, but if an LC-3 simulator is installed in the device, another method for an individual to utilize the program is by running the .asm file on the simulator.

* 1. Communications Interface

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

If the user does not have Java or an LC-3 simulator installed, they can utilize a web based simulator and run the .asm file on that manner. That being the case, that is when an internet connection would be required.

**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

Purpose: Calculate a student’s grade based on input data.

Inputs: Inputs come through the keyboard.

Processing:  The input is run through multiple mathematical calculations and for/if-else loops.

Outputs: The correct input will result in an output of a letter grade, two returned values (the smallest value input and the largest value input), and the percentage found. If incorrect data (letters and or invalid characters) are input, the user will be prompted to try again.

Data: Numerical values

Data Stored: Data is stored in registers.

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

**4.2.1** The application should be compatible with multiple OS’s due to its simplicity.

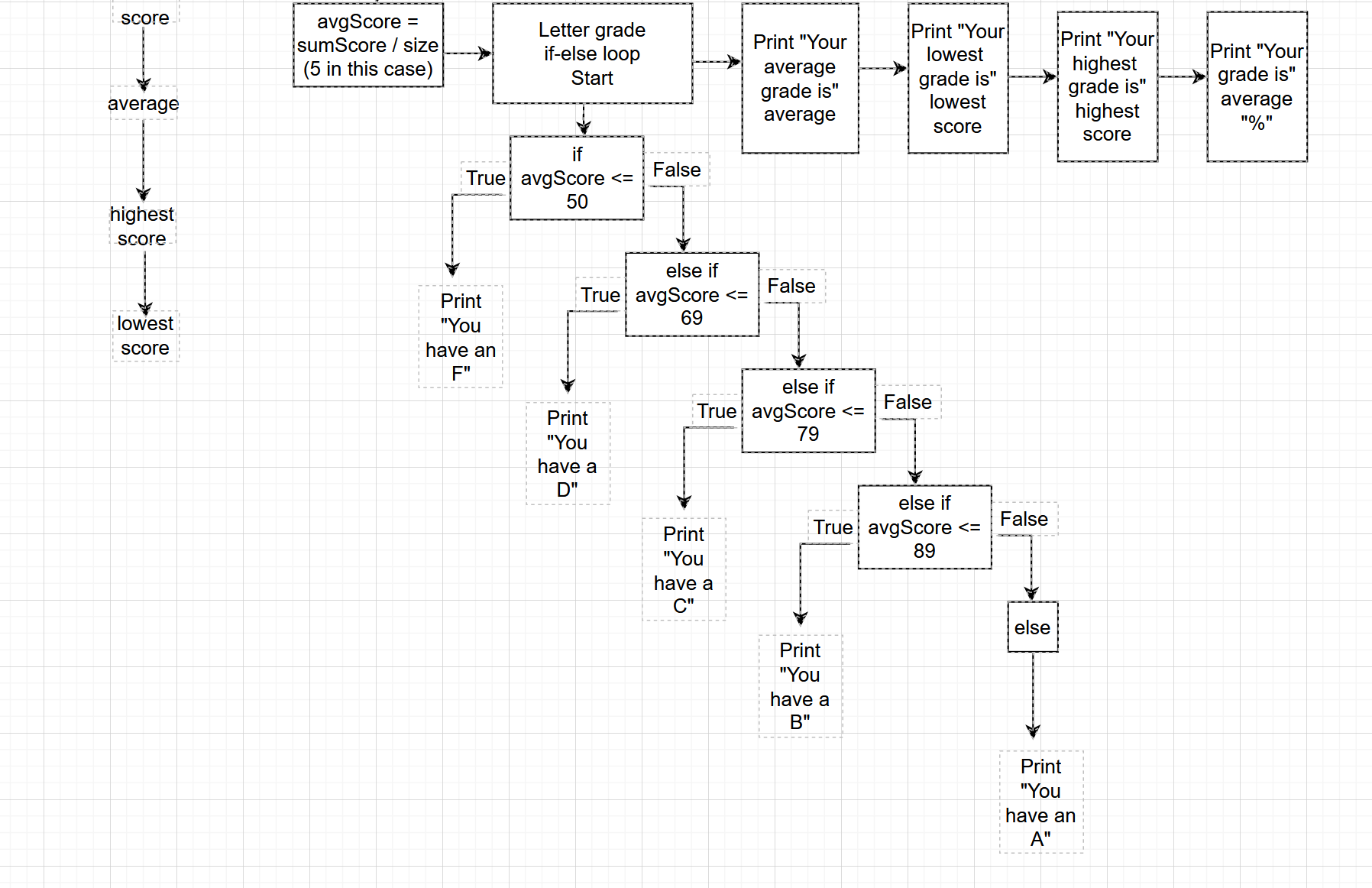
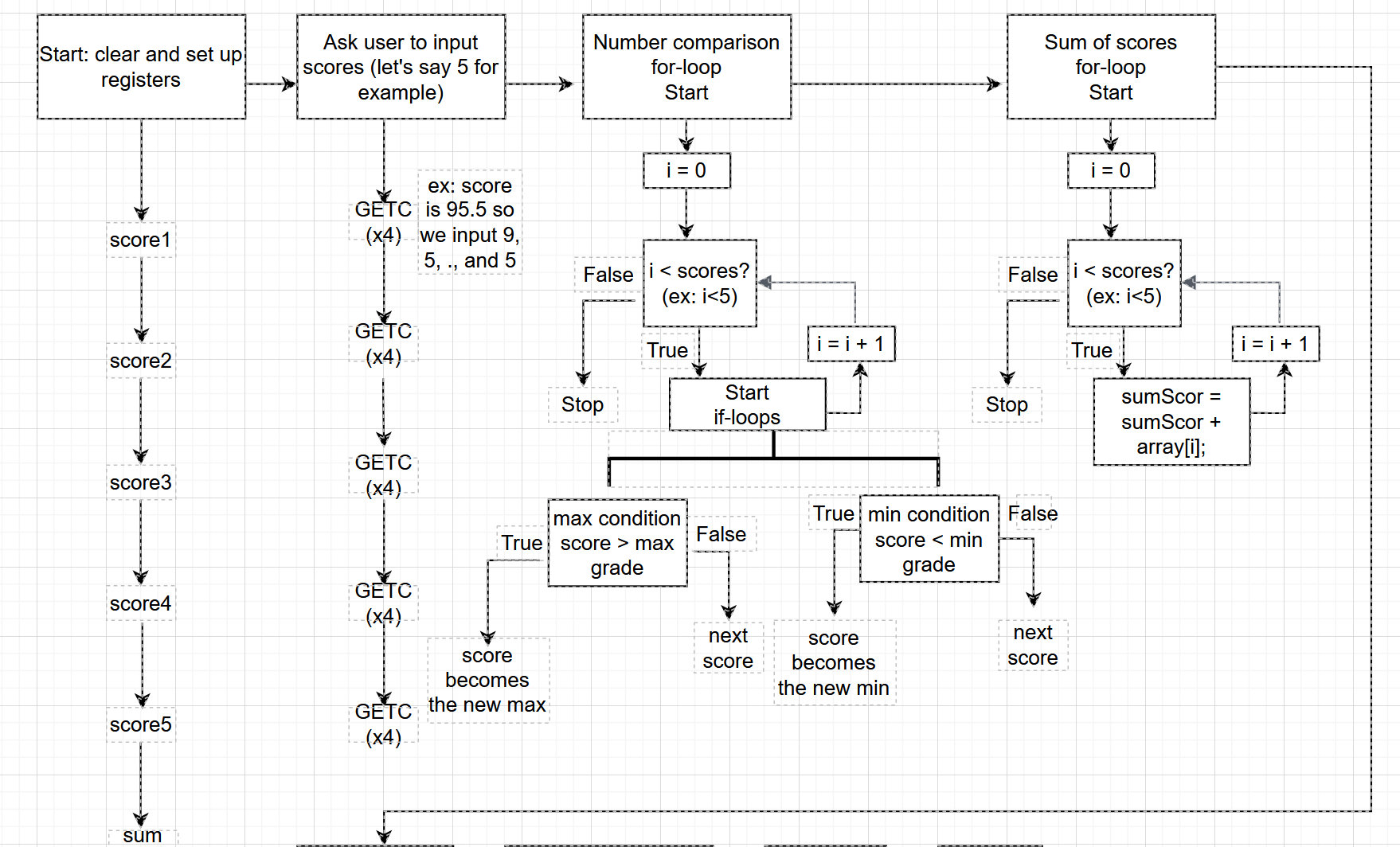
4.2.2 Since the application will solely be outputting words and calculations, the run time for each program should take 3-5 seconds after the user has input the scores.

4.2.3 The database should be scalable; it must have the capacity to hold a large array in the case that a student wants to check a whole semester of grades within the program.

**4.2.4** Error handling should be implemented and the application should be able to handle all run time errors.

**4.3 Flow Chart and Pseudocode.**

Flowchart:



Psuedocode:

Student inputs scores →

Program looks for the lowest score inputted →

Program looks for the highest score inputted→

Program adds each of the scores for the total →

Program divides the total by the size of the scores inputted to find the average →

Program runs through if-else loop →

    If 50 or below, letter grade is F

    If 69 or below, letter grade is D

    If 79 or below, letter grade is C

    If 89 or below, letter grade is B

    If 100 or below, letter grade is A

Program outputs average, lowest/highest scores, percentage, and letter grade

**5.1 Team Task Assignments**

**Students**

Allen Trujillo

* Code Compiler
* Core Programmer

Ryan Lane

* Advisor
* Tester
* Program Editor

Alejandra Andrade

* Flowchart/Pseudocode Designer
* Technical Writer

**Timeline**

* 5/3-9
  + **\*Team** Break down and assign tasks based on each team members’ strengths
* 5/10-16
  + **\*Documentation** Complete section 1 and 2
  + **\*Program** Go through previous labs and find relevant snippets for our purposes
  + **\*Program** Go through text and find relevant information for our purposes
  + **\*Program** Find online examples and resources of particular snippets
  + **\*Additional/Optional** Make alternative code (c++) that does the same tasks to visualize how to make accurate flowchart
* 5/17-23
  + **\*Documentation** Complete section 3 and 4
  + **\*Program** Start base of the code, and split the loops for other team members
  + **\*Additional/Optional** Make alternative code (c++) that does the same tasks to visualize how to make accurate flowchart
* 5/24-30
  + **\*Documentation** Finalize and turn in Documentation
  + **\*Program** Begin beta testing
  + **\*Program** Look for bugs in program
* 5/31-6/4
  + **\*Program** Finalize program
  + **\*Program** Continue testing
  + **\*Program** Make accessible through Java/Command Prompt
* 6/10
  + **\*Program** Begin turn in finalized LC-3 program
  + **\*Team** Turn in Evaluations