# **CIS11 Course Project Part 1: Documenting the Project**

Fill in the following areas (purple).

**Introduction**

**1.1** **Purpose**

The purpose of this program 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**

This program can be used by many students and professors to check their own grade out as well as the highest lowest and average score they received simply by inputting their test scores.

**1.3** **Product Scope**

What is the intention of this program?

The intention of this program is to take in 5 inputs of test scores, store them all to a location, convert the numbers to ascii, do the calculations for the min,max, and avg, and then convert it back to ascii for display purposes.

**1.4** **Reference**

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

Reference Project requirements and LC-3 specifications.

**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 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**

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

What other documents should be reviewed with this document?

Example:

1) CISData.doc

2) CISMigrationRefresh.doc

3) CISWorkflow.doc

4) CIS Process Model Diagrams (These require the SilverRun BPM viewer)

5) Scheduling system Class Schedule Book (CSB) File layout and description

**2. Overall Description**

**2.1 Product Perspective**

Primary program objectives

Taking the input of a person up to three digits. Finding the Minimum, Maximum and Medium of 5 test scores and finding a grade letter to print.

**2.2** **Product Functions**

**The overall description of functionality:**

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

1.List 5 grade scores in an array and using the JSR we calculate average, minimum, maximum, and average grade letters.

2.The average grade function sums all the grades together and divides it by 5.(no rounding in this program)

3.Minimum uses the first score and checks if it is less than second score and if it is then minimum score= Gradescore else it keeps looping testing the 3rd, 4th and 5th values

4.Maximum uses the first score and checks if it is greater than second score and if it is then Maximum score= Grade Score else it keeps looping testing the 3rd, 4th and 5th values

5. The Get Letter grade uses a bunch of if else statements to calculate whether the average of the grades is between the given values and it prints the appropriate letter

6. Then everything gets displayed in the GETDISPLAY function and prints to the console.

**Technical functionality**

A configurable toolkit of functions including:

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

GETAVERAGE\_GRADE

GETMINIMUM\_GRADE

GETMAXIMUM\_GRADE

GETLETTER\_GRADE

GETDISPLAY

2.3 **User Classes and Characteristics**

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

**1.Kenneth De La Cruz**

**Learning at Moreno Valley College**

**Documenting and flowchart.**

**2.Alyan Gillett**

**Learning at Moreno Valley College**

**documenting and creating the assembly language code.**

**3.Abdelrahman Sabbah**

**Learning at Moreno Valley College**

**Pseudocode, testing, github management**

**2.4** **Operating Environment**

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

The LC-3 program is locked to Windows only.

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

Note any constraints or limitations to the application.

Only being able to use numbers between 1-100. LC-3 program.

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

Note any dependencies

The test Values are required as well as a system to run it.

***3*. External Interface Requirements**

3.1 **User Interfaces**

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

The user must locate the user input at the bottom of the program and input values between 1-100.

3.2 **Hardware Interfaces**

Specify hardware interface – computer types? Terminal types?

Anything that can run a web browser to enter the code into.

3.3 **Software Interfaces**

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

Web browser or if the computer can download the LC-3 application.

3.4 Communications Interface

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

If the web browser, yes. If LC-3 application, no. Just a regular internet provider works.

**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)?**

Example: Grade Calculator

Purpose: get test scores and find the maximum, minimum and average scores then print them as well as a grade letter based on the average.

Inputs: Inputs are through the keyboard and mouse clicks.

Processing: looks at inputs and calculates accordingly through the subroutines.

Outputs: The maximum, minimum and average scores and the printed letter for the average.

**A. SUMOFSCORES/GETAVERAGE\_GRADE**

Purpose: take all 5 grades and find the average

Inputs: location where it is filled(keyboard and mouse clicks)

Processing: Add all 5 grades then divide it by 5

Outputs: prints the average of the grades on console

Data: User database

**B. GETMINIMUM\_GRADE**

Purpose: gets minimum out of all the scores

Inputs: location where it is filled(keyboard and mouse clicks)

Processing: takes a score and compares it to see if it is less than another score until it loops through all the scores to see which one is the smallest.

Outputs: prints the MINIMUM\_GRADE on console

Data: User database

**GETMAXIMUM\_GRADE**

Purpose: gets maximum out of all the scores

Inputs: location where it is filled(keyboard and mouse clicks)

Processing: takes a score and compares it to see if it is greater than another score until it loops through all the scores to see which one is the biggest.

Outputs: prints the MAXIMUM\_GRADE on console

Data: User database

**GETGRADE\_LETTER**

Purpose: prints letter grade based on average score.

Inputs:location where it is filled(keyboard and mouse clicks)

Processing: Checks to see where average score fits and if it does print the appropriate letter (0 – 50 = F, 60 – 69 = D, 70 – 79 = C, 80 – 89 = B, 90 – 100 = A)

Outputs: print letter grade

Data: User database.

**4.2 Performance requirements**

**What is the expected performance level of the program?**

1.The application should be accessible on any browser.

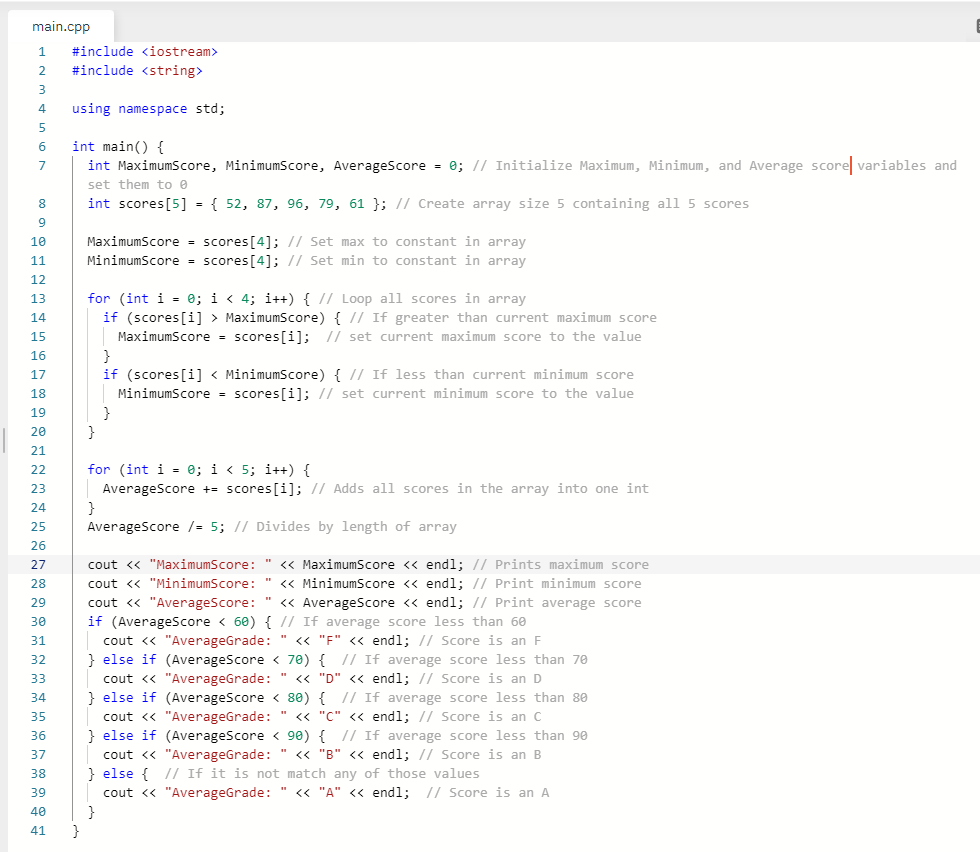
2. Since the program isn't long the estimated time for completion should be around 3-4 seconds after inputting scores.

3. The program as of now can take numbers 0-100 and still calculate it efficiently

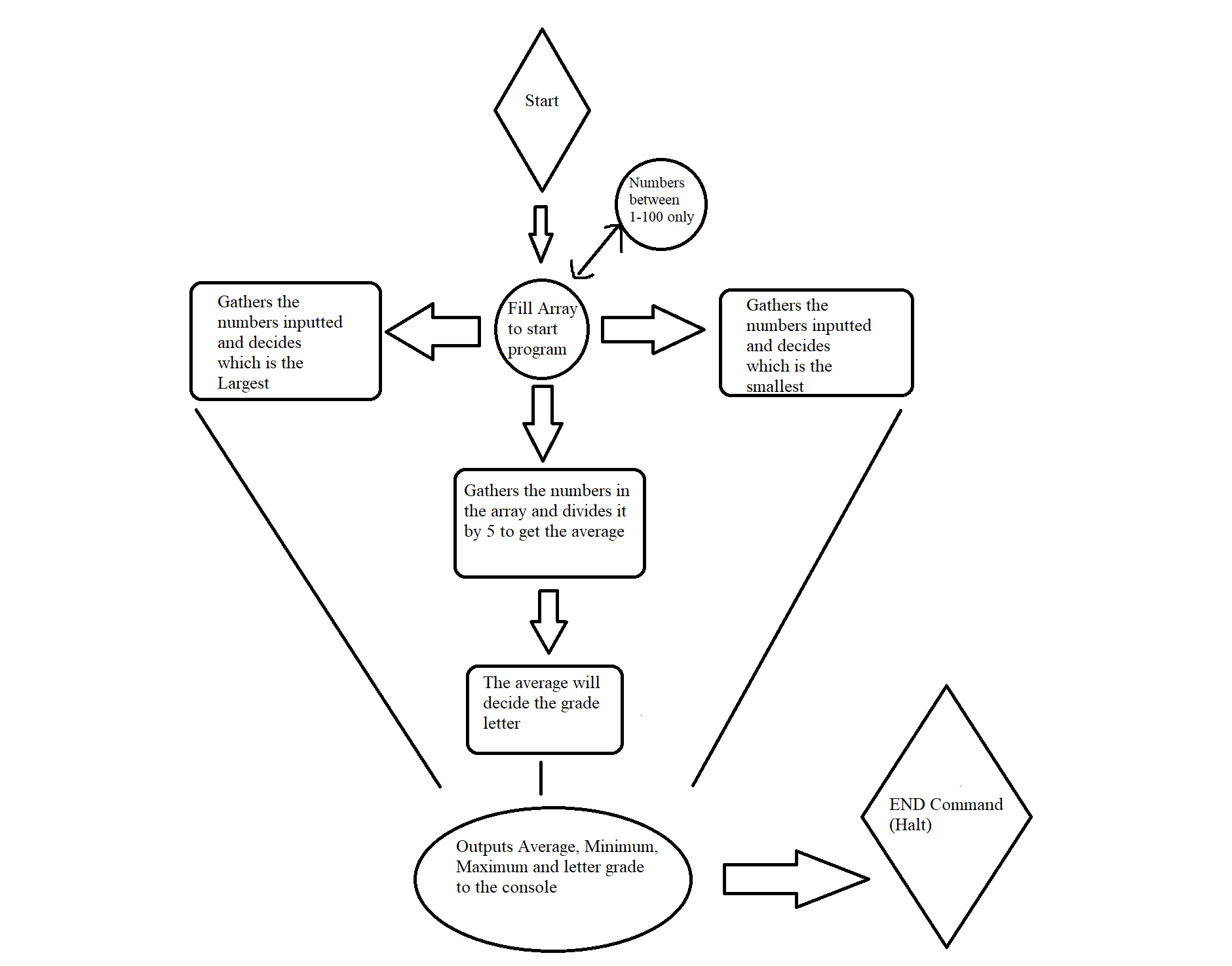
4. Error handling should be implemented and the application should be able to handle all runtime errors.

**4.3 Flow Chart and Pseudocode.**

**Pseudocode**

****

**Flow Chart**

****