# CIS11 Course Project Part 1: LC-3 Project

In this course project, we will create an LC-3 program that apply learned concepts in the course. This project is designed to capture the primary concepts of the course, which consist of computer architecture, processing and assembly programming. For this project, you will be working in teams, maximum 3 students per team. Each team will plan, design and build a successful LC-3 program. The program must fulfill the technical requirements. Refer to below project requirements.

In part 1, we will begin the planning and documentation process (100 points)

## Tasks:

1. **Review the Project Program Requirements (see below).**
2. **Derive a list of program objectives (include list in Project Documentation file): What are the program tasks (by order)?**
3. **Use the listed objectives to construct pseudocode and flowchart for the program, which will be part of the documentation.**
4. **Begin documentation to address project purpose, objectives and outcome. See Project Documentation file.**

## Project Program Requirements:

### Option B: Test Score Calculator

**Program**

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 score 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 52 | 87 | 96 | 79 | 61 |

Building and testing the program is for Project Part 3 (Total 300 points).