**PROJECT REQUIREMENTS AND INSTRUCTIONS:**

1. REQUIREMENTS
   1. The program must use the following programming concepts:
      1. Functions
      2. Arrays
      3. Structures
      4. Pointers and/or Linked List
      5. File Manipulation
   2. The program must use an external file (.txt) as the storage of all the records that are added to the program.
   3. The program must have a Main Menu wherein the user can select the operation he wants. Operations must be:
      1. Add new record
      2. Search record
      3. Delete record
      4. Display records
   4. The program must display the name of your group and the names of the members of the group when the user exits from the program.
   5. The program must have basic error trapping feature, like when a user enters an invalid input, or invalid form of data.
2. INSTRUCTIONS
   1. Create a Student Information program.
   2. The program will ask the user what he wants to do by providing him a menu that shows the operations that the program can do. Example:

Welcome to Group Pr0gr@mm@z Student Information System

What do you want to do?

1. Add New Record
2. Search Record
3. Display All Records
4. Display Specific Record
5. Delete Record
6. Exit

Please type your selection: \_\_

* 1. When the user selects an operation from the list, the program will execute the selected operation.
  2. If the user inputs a number that is not in the menu, e.g. 7, the program will tell the user that he entered a wrong input and will ask the user to enter another selection.
  3. The program must store the following:
     1. Student ID Number
     2. Full Name
     3. Birthday
     4. Address
     5. Gender
     6. Degree Program
     7. Year Level
  4. Your program can have one or more ways of searching for a record such as searching by name, or by ID Number.
  5. Create a video presentation which shows how your program is used and how it works. Any member or all members of the group can appear in the video. (This video presentation will be upload to your Google Drive or to any video streaming platform online).



Department of Information Technology

CCS0007 – COMPUTER PROGRAMMING 2

[SECTION]

Final Project

Grade

Submitted by:

PRINTED MEMBER PICTURE

LN, FN MI

PRINTED MEMBER PICTURE

LN, FN MI

PRINTED MEMBER PICTURE

LN, FN MI

PRINTED MEMBER PICTURE

LN, FN MI

|  |
| --- |
| Submitted to:  [Name of Course Adviser] |

**DOCUMENT CONTENTS:**

1. **INTRODUCTION**

Provide some statement about your project or computer program.

1. **OBJECTIVES**

State the main objective of the project.

1. **PROGRAM SCREEN SHOTS**

Capture every screen of your running program and provide some explanation what is in the screen shot. See example below…

PROGRAM

**Figure 1. Main Screen**

Figure 1 shows ... (*provide some statement pertaining to the figure*)

PROGRAM DEMONSTRATION VIDEO: \_\_\_\_\_\_\_(paste video URL here)\_\_\_\_\_\_\_\_\_\_

1. **SOURCE CODE**

Paste the program codes with proper comments for each module and some explanation statements. **Use Courier New as your font for your codes, and Arial for the explanation.**

1. **SUMMARY AND DISCUSSION**

Write a summary about the program, the program codes, and the things that you have learned from doing this program and ultimately from this course.

1. **RUBRICS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRAIT | Exceptional | Acceptable | Amateur | Unsatisfactory |
| Specifications  (20) | The program works and meets all of the specifications. | The program works and produces the correct results and displays them correctly. It also meets most of the other specifications. | The program produces correct results but does not display them correctly. | The program is producing incorrect results. |
| Readability  (20) | The code is exceptionally well organized and very easy to follow. | The code is fairly easy to read. | The code is readable only by someone who knows what it is supposed to be doing. | The code is poorly organized and very difficult to read. |
| Reusability  (20) | The code could be reused as a whole or each routine could be reused. | Most of the code could be reused in other programs. | Some parts of the code could be reused in other programs. | The code is not organized for reusability. |
| Documentation  (30) | The documentation is well written and clearly explains what the code is accomplishing and how. | The documentation consists of embedded comment and some simple header documentation that is somewhat useful in understanding the code. | The documentation is simply comments embedded in the code with some simple header comments separating routines. | The documentation is simply comments embedded in the code and does not help the reader understand the code. |
| Efficiency  (10) | The code is extremely efficient without sacrificing readability and understanding. | The code is fairly efficient without sacrificing readability and understanding. | The code is brute force and unnecessarily long. | The code is huge and appears to be patched together. |