# Technological Institute of the Philippines

**Quezon City**

**S.Y 2019- 2020**

**Automatic Grading System for Junior High School and Senior High School Teachers Using the C Language**

**Kristian John Q. Baturiano**

**Jomar Angelo U. Rempillo**

**CPE11FB1**

**Instructor:**

**Jonathan Tylar**

**Submission Date:**

**September 15, 2019**

# TABLE OF CONTENTS

[TABLE OF CONTENTS 2](#_Toc19463078)

[CHAPTER 1: PROJECT BACKGROUND 3](#_Toc19463079)

[1.1 The Project 3](#_Toc19463080)

[1.2 Objectives 3](#_Toc19463081)

[1.3 Scope and Delimitation 4](#_Toc19463082)

[CHAPTER 2: PROJECT DESIGN 5](#_Toc19463083)

[2.1 Project Flow Chart 5](#_Toc19463084)

[2.2 Algorithm 6](#_Toc19463085)

[CHAPTER 3: FINAL DESIGN 7](#_Toc19463086)

[3.1 Screenshot of the Process of the Program 7](#_Toc19463087)

[3.2 Compressed Copy of the Code 9](#_Toc19463088)

[INDICES 14](#_Toc19463089)

[References 14](#_Toc19463090)

[Curriculum Vitae 15](#_Toc19463091)

[Evidence of Working in Groups 17](#_Toc19463092)

# CHAPTER 1

**PROJECT BACKGROUND**

## 1.1 The Project

The San Juan National High School (SJNHS) is a public secondary school in the City of San Juan. Located in P. Narciso St., Brgy. Corazon de Jesus, San Juan City, it provides free secondary education for the people of the municipality and neighboring towns with primary aim on secondary basic education. Liberty D. Quirino, the project team’s client, a junior high school teacher working at the said school teaching mainly Pre-Calculus and Trigonometry, said that she is having a hard time keeping track of the grades of all the students under her. The problem includes the enormous weight of the paper she needs to keep track of all the student and their grades and her desire for an efficient system manifested into this project. She wishes to have a system that encodes the grade of her students according to three categories (the three categories being Performance Tasks, Written Works, and Periodical Exam) by grading period and will automatically compute the final grade of the grading period and record the data that has been entered. This project, which aims to be able to automate the process of encoding grades and be able to store all entries where they can be retrieved whenever necessary, serves as a solution to our client’s requirement for such a system.

## 1.2 Objectives

The objectives of the project are as follows:

1. To create a program that encodes and stores the grade of students according to the three main categories being:
2. Performance Tasks
3. Written Works
4. Periodical Exam
5. To be able to edit and delete any record according to the user’s needs.
6. To use the program to display the grade of the student in order and display the total grade when needed

## 1.3 Scope and Delimitation

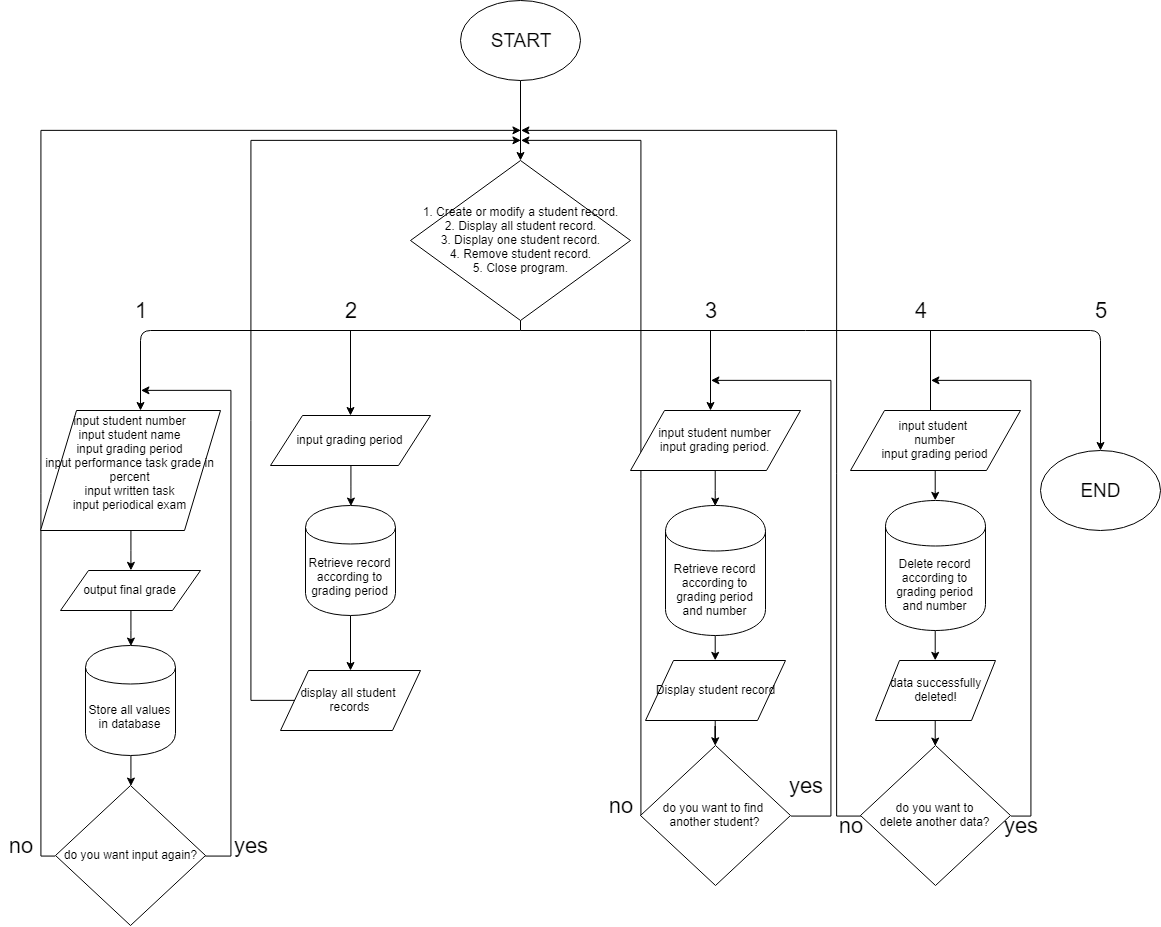
The project aims to create a program that receives and stores the grade of a student per category, and displays the record of the student for each category and the total grade. The program’s main purpose is to store the grades of the students and make the calculation and tracking of the grades easier. This program itself is written exclusively in the C language. The database language used for the project is strictly MySQL. The UI (user interface) is limited to what the C language is capable of. The program cannot display the specific names of each activity/quiz/exam that are encoded in the program.

# CHAPTER 2

**PROJECT DESIGN**

## 2.1 Project Flow Chart

The figure below shows the graphical way of the program on how it should run.

****

**Figure 1.** The flowchart detailing the flow of the program

## 

## 2.2 Algorithm

Step 1: Choose the number of the task that you want to perform. (1) Create or modify a student record. (2) Display all student record. (3) Display one student record. (4) Remove student record. (5) Close program.

Step 2: If 1 is chosen, enter the class number, student’s name, the grading period, grade in performance task, written works and periodical exam. All inputs will be stored in the database.

Step 3: If you want to create or update another record, press 1, if not go to step 1.

Step 4: If 2 is chosen, the program will retrieve the data from the database and display all the student record. And it will automatically display the main menu after.

step 5: If 3 is chosen, enter the class number and enter the grading period of the desired. The program will retrieve the data from the database and display the desired record.

Step 6: If you want to find another specific record, press 1, if not go to step 1.

Step 7: If 4 is chosen, enter the class number of the student and enter the grading period of the record that you want to delete.

Step 8: If you want to delete another record, press 1 and if not go to step 1.

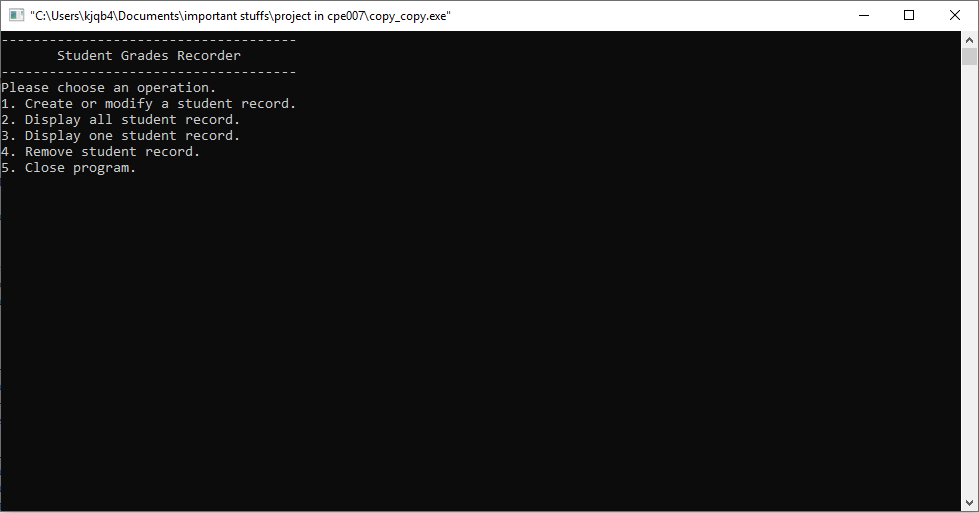
Step 9: If 5 is chosen, the program will terminate.

# CHAPTER 3

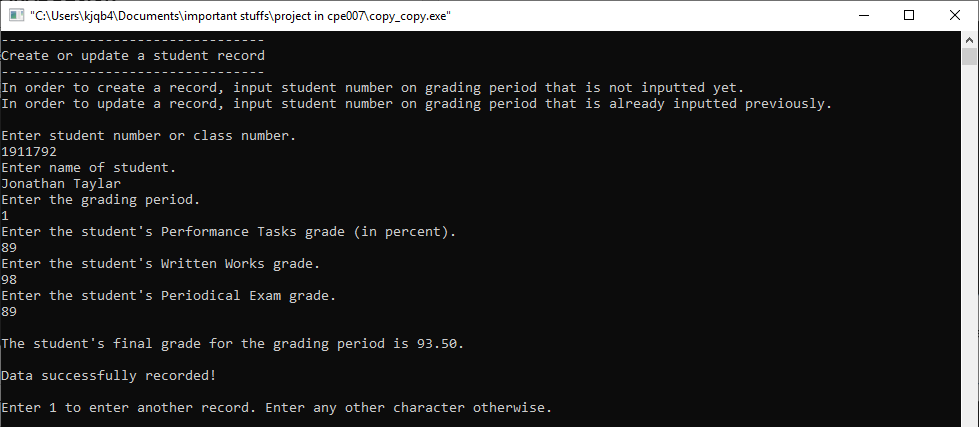
**FINAL DESIGN**

## 3.1 Screenshot of the Process of the Program

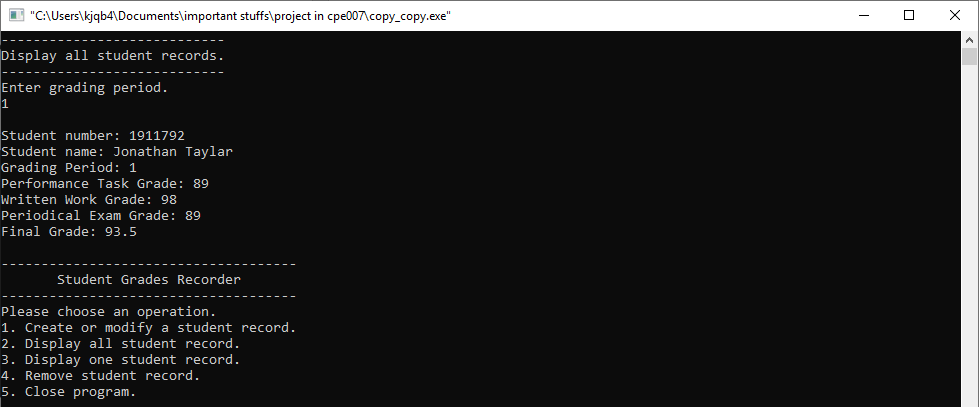
Here you are going to choose the task that you want to perform.



**Figure 1.** Program Main Menu

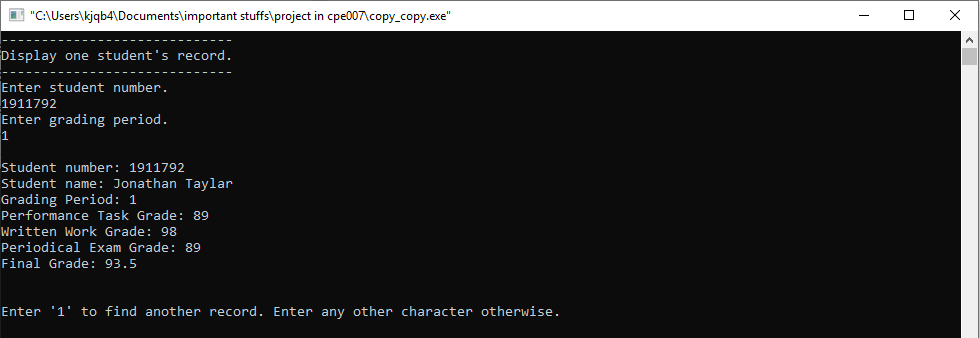
Option 1 is “Create or modify a student record”, where you’ll enter the student number, student’s name, the grading period, grade in performance task, grade in written works and the grade in periodical exam. **Figure 2.** Create or update student record

Option 2 is “Display all student records”, where you’re going to see all the students record for one grading period.



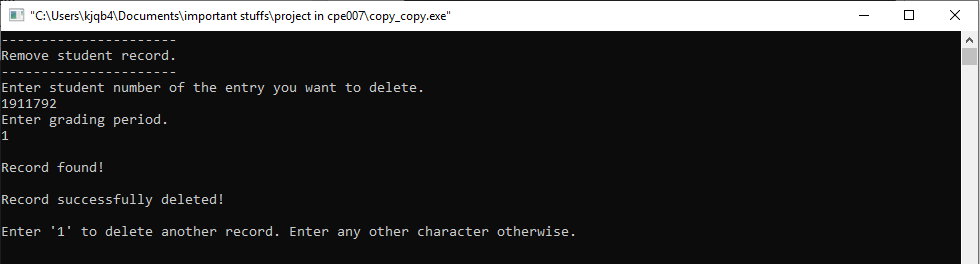
**Figure 3.** Display all student record for one grading period

Option 3 is “Display one student record”, where you’re going to enter the number and the grading period of the record that you want to see.



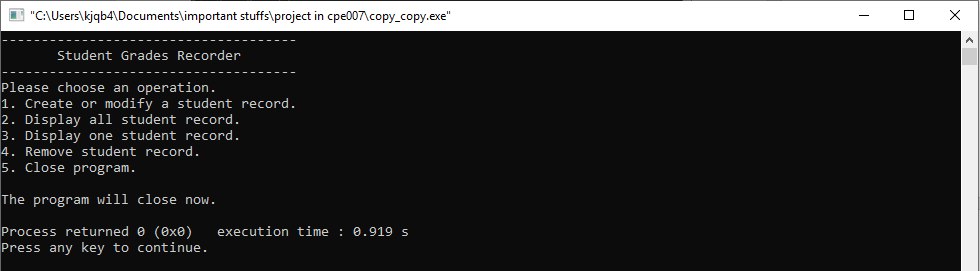
**Figure 4.** Display one specific student record.

Option 4 is “Remove student record”, where you’re going to enter the number and the grading period of the record that you want to delete.



**Figure 5.** Remove one specific student record.

Option 5 is “Close program”, where the program will automatically terminate.



**Figure 6.** Terminating the program

## 3.2 Compressed Copy of the Code

#include <windows.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <mysql.h>

#include <conio.h>

void creates();

void reads();

void finds();

void updates();

void deletes();

void user\_manual();

//for debugging purposes

void finish\_with\_error(MYSQL \*con) {

fprintf(stderr, "%s\n", mysql\_error(con));

mysql\_close(con);

exit(1);

}

int main() {

//initializes MySQL

MYSQL \*con = mysql\_init(NULL);

if (con == NULL) {

finish\_with\_error(con);

}

//connects to a database that corresponds to the details in mysql\_real\_connect

//if being used at another computer, export and import cpe\_project database then change username and password in mysql\_real\_connect function according to settings of other computer

if (mysql\_real\_connect(con, "localhost", "root", "8700", "cpe\_project", 0, NULL, 0) == NULL) {

finish\_with\_error(con);

}

int x;

while (1) {

printf("-------------------------------------\n");

printf(" Student Grades Recorder \n");

printf("-------------------------------------\n");

printf("Please choose an operation.\n");

printf("1. Create or modify a student record.\n");

printf("2. Display all student record.\n");

printf("3. Display one student record.\n");

printf("4. Remove student record.\n");

printf("5. Close program.\n");

x = getch();

switch (x) {

case '1':

creates(con);

break;

case '2':

reads(con);

break;

case '3':

finds(con);

break;

case '4':

deletes(con);

break;

case '5':

mysql\_close(con);

printf("\nThe program will close now.\n");

exit(0);

default:

printf("\nIncorrect input.\n\n");

break;

}

}

}

//function that creates or edits an entry in the database

void creates(MYSQL \*con) {

int y = '1';

char name[50];

int student\_num, grading\_period;

float pt\_grade, ww\_grade, pe\_grade, final\_grade;

while (y == '1') {

system("CLS");

printf("---------------------------------\n");

printf("Create or update a student record\n");

printf("---------------------------------\n");

printf("In order to create a record, input student number on grading period that is not inputted yet.\n");

printf("In order to update a record, input student number on grading period that is already inputted previously.\n");

printf("\nEnter student number or class number.\n");

fflush(stdin);

scanf("%d", &student\_num);

printf("Enter name of student.\n");

fflush(stdin); //used because of fgets

fgets(name, 50, stdin);

name[strcspn(name, "\n")] = 0;

printf("Enter the grading period.\n");

scanf("%d", &grading\_period);

printf("Enter the student's Performance Tasks grade (in percent).\n");

scanf("%f", &pt\_grade);

printf("Enter the student's Written Works grade.\n");

scanf("%f", &ww\_grade);

printf("Enter the student's Periodical Exam grade.\n");

scanf("%f", &pe\_grade);

final\_grade = (pt\_grade\*0.3)+(ww\_grade\*0.5)+(pe\_grade\*0.2); //grades are multiplied according to how much they comprise the final grade, the percentage is taken from the client herself

printf("\nThe student's final grade for the grading period is %0.2f.\n", final\_grade);

char buf[1024]={}; //the array that will hold the query

//the long query is due to it being a create and update query. if it doesn't find a duplicate, which is based on the student number, it will enter the data inputted by the user, otherwise it will overwrite the duplicate

char query[]={"INSERT INTO grade\_%d(student\_num,name,pt\_grade,ww\_grade,pe\_grade,final\_grade) VALUES('%d','%s','%f','%f', '%f', '%f') ON DUPLICATE KEY UPDATE student\_num=VALUES(student\_num),name=VALUES(name),pt\_grade=VALUES(pt\_grade),ww\_grade=VALUES(ww\_grade),pe\_grade=VALUES(pe\_grade), final\_grade=VALUES(final\_grade)"};

sprintf(buf,query,grading\_period,student\_num,name,pt\_grade,ww\_grade,pe\_grade,final\_grade); //using sprintf, a query can be written with user input values from C

if (mysql\_query(con,buf)) {

finish\_with\_error(con);

}

else {

printf("\nData successfully recorded!\n");

}

printf("\nEnter 1 to enter another record. Enter any other character otherwise.\n");

y = getch();

}

}

//function that reads all entries for one grading period

void reads(MYSQL \*con) {

system("CLS");

printf("----------------------------\n");

printf("Display all student records.\n");

printf("----------------------------\n");

printf("Enter grading period.\n");

int period;

scanf("%d", &period);

char buf[1024]={};

sprintf(buf,"SELECT \* FROM grade\_%d",period);//retrieves all values from rows that corresponds to grading period

if (mysql\_query(con,buf)) {

finish\_with\_error(con);

}

MYSQL\_ROW row;

MYSQL\_RES \*res = mysql\_store\_result(con);

while (row=mysql\_fetch\_row(res)) {//this will print out the data it receives as long as there are rows detected

printf("\n");

printf("Student number: %s\n",row[1]);

printf("Student name: %s\n",row[2]);

printf("Grading Period: %d\n",period);

printf("Performance Task Grade: %s\n",row[3]);

printf("Written Work Grade: %s\n",row[4]);

printf("Periodical Exam Grade: %s\n",row[5]);

printf("Final Grade: %s\n",row[6]);

printf("\n");

}

}

//function that reads one specific record that matches the user inputted student number and grading period

void finds(MYSQL \*con) {

system("CLS");

int y='1';

printf("-----------------------------\n");

printf("Display one student's record.\n");

printf("-----------------------------\n");

while (y=='1') {

printf("Enter student number.\n");

int student;

scanf("%d", &student);

printf("Enter grading period.\n");

int period;

scanf("%d", &period);

char buf[1024]={};

sprintf(buf,"SELECT \* FROM grade\_%d WHERE student\_num=%d", period, student); //retrieves all values from row that corresponds to student number and grading period

if (mysql\_query(con,buf)) {

finish\_with\_error(con);

}

else {

MYSQL\_ROW row;

MYSQL\_RES \*res = mysql\_store\_result(con);

while(row=mysql\_fetch\_row(res)) {

printf("\n");

printf("Student number: %s\n",row[1]);

printf("Student name: %s\n",row[2]);

printf("Grading Period: %d\n",period);

printf("Performance Task Grade: %s\n",row[3]);

printf("Written Work Grade: %s\n",row[4]);

printf("Periodical Exam Grade: %s\n",row[5]);

printf("Final Grade: %s\n",row[6]);

printf("\n");

}

}

printf("\nEnter '1' to find another record. Enter any other character otherwise.\n");

y = getch();

}

}

////function that deletes one specific record that matches the user inputted student number and grading period

void deletes(MYSQL \*con) {

system("CLS");

int y='1';

while (y=='1') {

printf("----------------------\n");

printf("Remove student record.\n");

printf("----------------------\n");

printf("Enter student number of the entry you want to delete.\n");

int student;

scanf("%d", &student);

printf("Enter grading period.\n");

int period;

scanf("%d", &period);

char buf[1024]={};

sprintf(buf,"SELECT \* FROM grade\_%d WHERE student\_num=%d",period,student);

if (mysql\_query(con,buf)) {

finish\_with\_error(con);

}

else {

MYSQL\_ROW row;

MYSQL\_RES \*res = mysql\_store\_result(con);

if ((row=mysql\_fetch\_row(res))) {

printf("\nRecord found!\n");

char buf\_delete[1024]={};

char query\_delete[]={"DELETE FROM grade\_%d WHERE student\_num=%d"}; //

sprintf(buf\_delete,query\_delete,period,student);

if (mysql\_query(con,buf\_delete)) {

finish\_with\_error(con);

}

else {

printf("\nRecord successfully deleted!\n");

}

}

else {

printf("\nNo such record found!\n");

}

}

printf("\nEnter '1' to delete another record. Enter any other character otherwise.\n\n");

y = getch();

}

}

# INDICES

## References

J. Bodnar, “MySQL C API programming tutorial”, April, 2013. [Online]. Available: <http://zetcode.com/db/mysqlc/?fbclid=IwAR34wPycGRIDvERsc4q8XdHrnji31MZYaM9afZiKjhNuMD62TRXPVgPvFAw>. [Accessed: Sept-14-2019].

J. Leffler, “Removing trailing newline character from fgets() input,” *Stack Overflow*, 14-Mar-2015. [Online]. Available: https://stackoverflow.com/questions/2693776/removing-trailing-newline-character-from-fgets-input. [Accessed: 14-Sep-2019].

“On Duplicate Key Update same as insert,” *Stack Overflow*, 17-Jan-2017. [Online]. Available: https://stackoverflow.com/questions/14383503/on-duplicate-key-update-same-as-insert. [Accessed: 14-Sep-2019].

“Fetching rows in a MySQL database table using MySQL C API and C ,” *Stack Overflow*, 28-Jul-2009. [Online]. Available: <https://stackoverflow.com/questions/1194453/fetching-rows-in-a-mysql-database-table-using-mysql-c-api-and-c>. [Accessed: Sept-14-2019]

“MySQL 5.7 Reference Manual :: 27.8 MySQL C API,” MySQL. [Online]. Available: https://dev.mysql.com/doc/refman/5.7/en/c-api.html?fbclid=IwAR2zoOLCWlPm-1UnzaVqMZOY7Zh0n8KPpSBiEgIySGCaElROibP7BO8ckwY. [Accessed: 14-Sep-2019].

## Curriculum Vitae

**Personal Data**

**Name:** Kristian John Q. Baturiano

**Nickname:** Anjo

**Gender:** Male

**Status:** Single

**Birthday:** February 14, 2001

**Home Address:** 7 B. Padilla St., Brgy. Sta. Lucia,

San Juan City

**Email Address:** kjqb4929@gmail.com

**Family Background**

**Mother:** Elisa Q. Baturiano **Father:** Norman O. Baturiano

**Educational Background**

**Tertiary**  Technological Institute of The Philippines

**Level:**

**Secondary** San Juan National High School

**Level:**

**Primary** Kabayanan Elementary School

**¬Level:**

**Personal Data**

**Name:** Jomar Angelo U. Rempillo

**Nickname:** Mar

**Gender:** Male

**Status:** Single

**Birthday:** November 23, 2000

**Home Address:** Blk 9 Lot 11 Steelhomes Subdivision,

Brgy. San Luis, AntipoloCity

**Email Address:** Jo23mar112300@gmail.com

**Family Background**

**Mother:** Josephine R. Bautista **Father:** Mar R. Bautista

**Educational Background**

**Tertiary** Technological Institute of The Philippines

**Level:**

**Secondary** Antipolo National High School

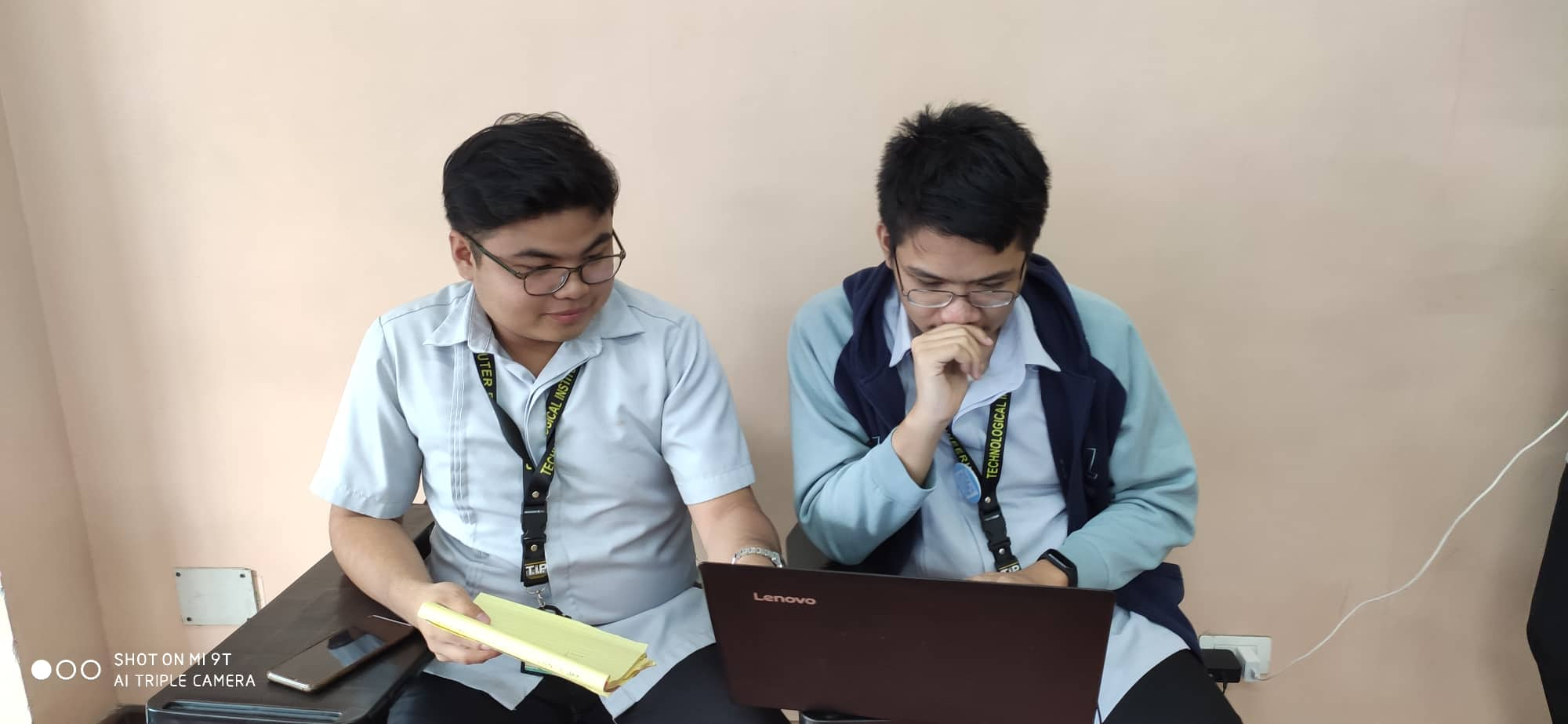
**Level:**

**Primary** Sta.Cruz Elementary School

**­Level:**

## Evidence of Working in Groups

****

****