<u>Computer Programming 2 Laboratory</u>

Project Documentation

Mapua University
School of Information Technology

MAPUA INSTITUTE OF TECHNOLOGY SCHOOL RECORDS

By
PASTORAL, Lorenzo Troy P.

In partial fulfillment of the Requirements of the Coutser
CS127-8L

March 26, 2019
3rd Quarter / SY 2019 - 2020

Table of Contents

INTRODUCTION

The function of the program is to simulate a school records system for both students and teachers with an encrypted log-in system. The program enables the user to interact with the various functions of the program— which is to create, display, search, show, update and delete records. The program will present these in categories for both students and teachers. The program has also a built-in calculator that will calculate the GWA of a student. The system priorities the main organisation of data that is inputted by the user. The program first prompts the user to log-in as admin in order to manipulate the functions within the program. Once the log-in was the successful, the user can choose whether to manipulate the functions within the student or teacher category. To note, both categories possess similar functions.

The student records category has a built-in structures system, which serves as a template to document the data; the elements within the structure are such as name, course, email, scholarship validity, school year, phone number, student number, age, GWA as well as record index. The student records menu has the functions such as upload, delete, update, search, show, export and calculate grades. These are to help manipulate the data inputted by the user. All these functions attempt to simulate what a records system might utilise. The teacher records category also has a built-in structures system, which serves as a template to document the data; the elements within the structure are such as name, course, email, scholarship validity, school year, phone number, student number, age, GWA as well as record index. The student records menu has the functions such as upload, delete, update, search, show, export and calculate grades. These are to help manipulate the data inputted by the user. The objective is to manipulate data coming from user as well as give the user the ability to export all the given data onto the local drive. This function can be found under the export function, which makes the function useful as it can record data which can be used as future reference.

DEVELOPMENT

The entirety of the functions mainly utilises mainly the concept of structures and arrays. The program is based of a iteration structure that will allow the user to create as much input values as the program permits (1000 max for this given program). String manipulations, loops, arrays, structures, and validity checking are the tools that are used to further the program.

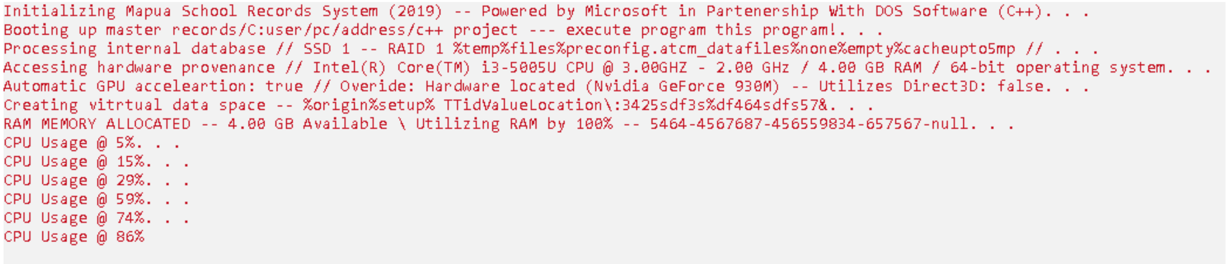
Another notable part of this program is the utilisation of file streaming. This allows the user to export data to an external file utilising via the **fstream library.** The input stream also allows the program to check the validity of the programming by referencing the input of the user to the external program. If both data match, this enables the user to run the program, if not, it will prompt the user to repeat the log-in sequence. Switch cases and do-while loops help authenticate and validate the input of the user.

There are also many instances of indexing loops and structures that help the user traverse through the subscript position of each structured array (for students and teachers).

Try and catch blocks are also utilised to validate the range of the index ID that is input by the user. The main logic relies on the program creating to discrete structured arrays / loops for both student and teacher records. What ties the program together is the amalgamation of the numerous concepts— arrays, loops, structures, file streaming, and c-strings— that help creates a harmonious program.

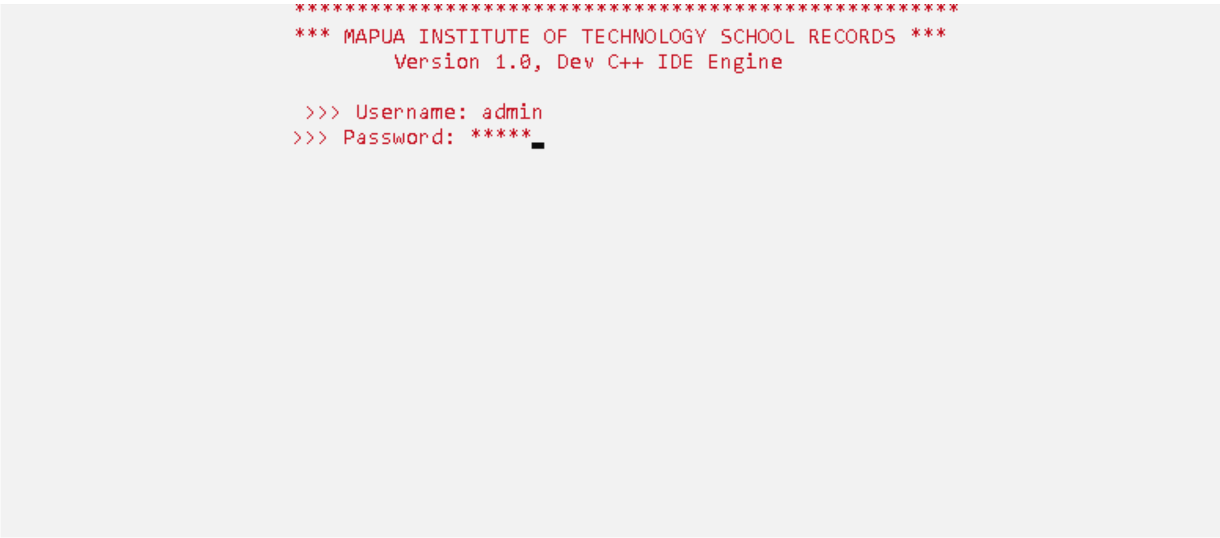
CODE SNIPPETS

A custom initializion boot-up window appears before the actual program is executed. (Note: This is more for show, no specific function)



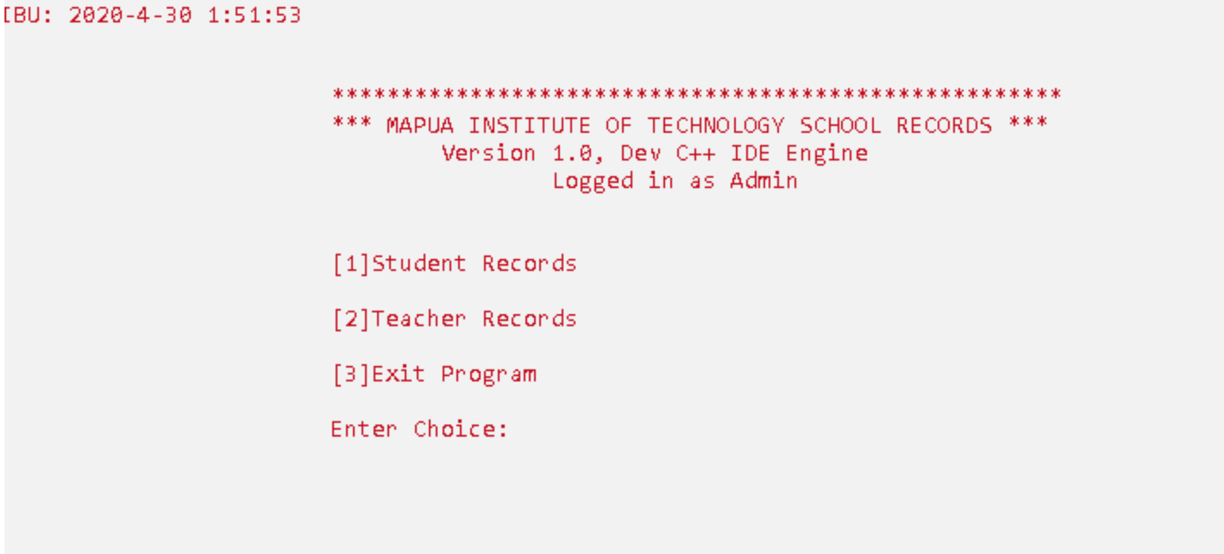
```
Initializing Mapua School Records System (2019) -- Powered by Microsoft in Partenership With DOS Software (C++). . .
Booting up master records/C:user/pc/address/c++ project --- execute program this program!. . .
Processing internal database // SSD 1 -- RAID 1 %temp%files%preconfig.atcm_datafiles%none%empty%cacheupto5mp // . . .
Accessing hardware provenance // Intel(R) Core(TM) i3-5005U CPU @ 3.00GHZ - 2.00 GHz / 4.00 GB RAM / 64-bit operating system. . .
Automatic GPU acceleartion: true // Overide: Hardware located (Nvidia GeForce 930M) -- Utilizes Direct3D: false. . .
Creating vitrtual data space -- %origin%setup% TTidValueLocation\:3425sdf3s%df464sdfs57&. . .
RAM MEMORY ALLOCATED -- 4.00 GB Available \ Utilizing RAM by 100% -- 5464-4567687-456559834-657567-null. . .
CPU Usage @ 5%. . .
CPU Usage @ 15%. . .
CPU Usage @ 29%. . .
CPU Usage @ 59%. . .
CPU Usage @ 74%. . .
CPU Usage @ 86%
```

User is asked to input the valid username and password (that being admin for both respectively).



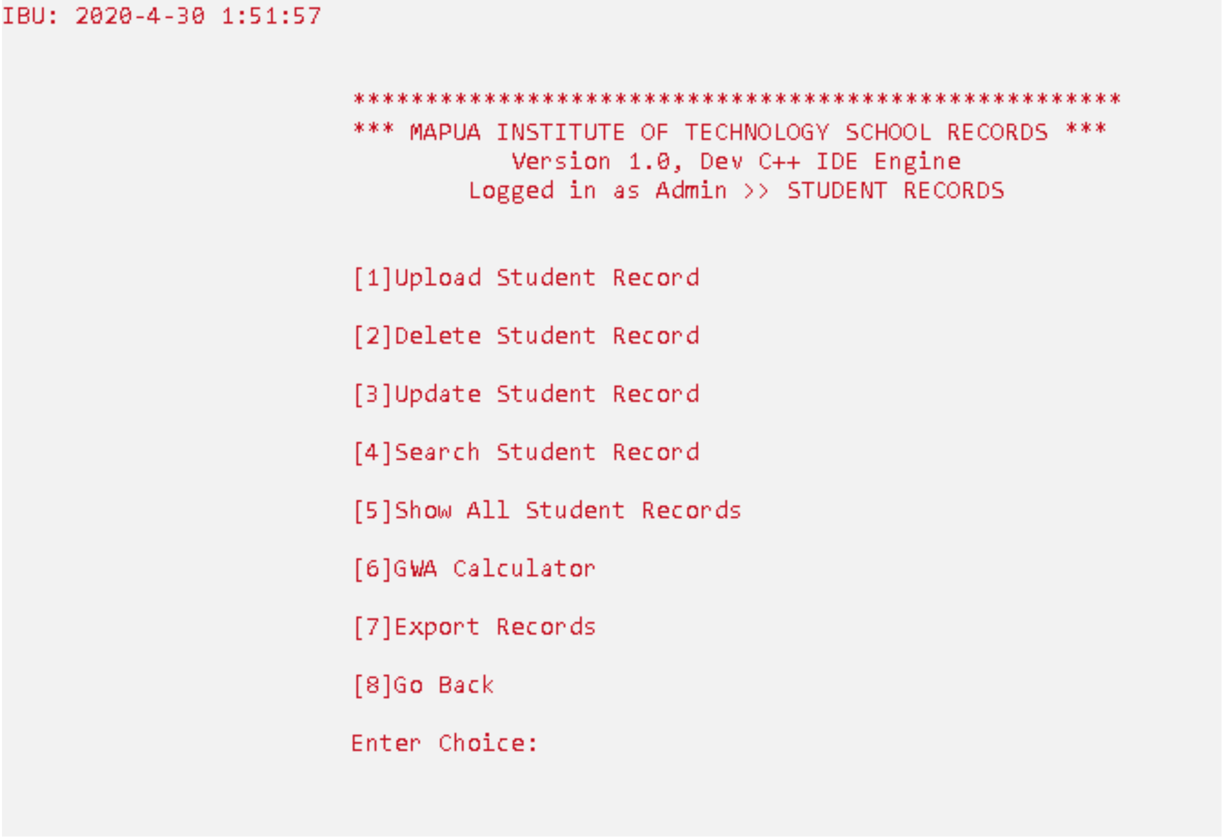
```
******************************************************
*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL RECORDS ***
        Version 1.0, Dev C++ IDE Engine

 >>> Username: admin
>>> Password: *****
```

User is then asked to select one of the three options: student menu, teacher menu or exit program. The user then needs
to input the corresponding choice.



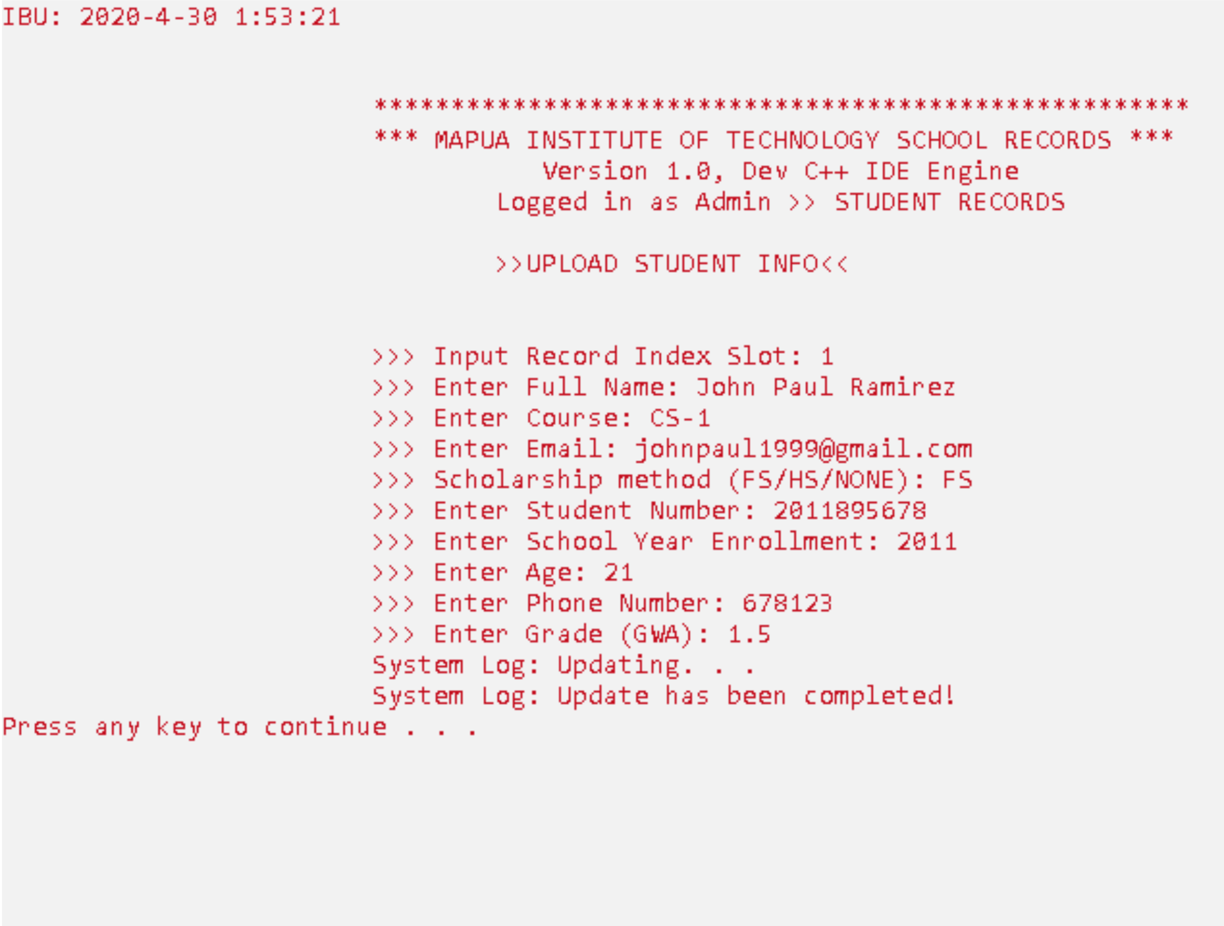
```
[BU: 2020-4-30 1:51:53

******************************************************
*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL RECORDS ***
        Version 1.0, Dev C++ IDE Engine
                Logged in as Admin


[1]Student Records

[2]Teacher Records

[3]Exit Program

Enter Choice:
```

Here is a snippet of the student menu (Note: Both student and teacher menus have similar functions). The user is prompted
to enter in their choice with the given options.



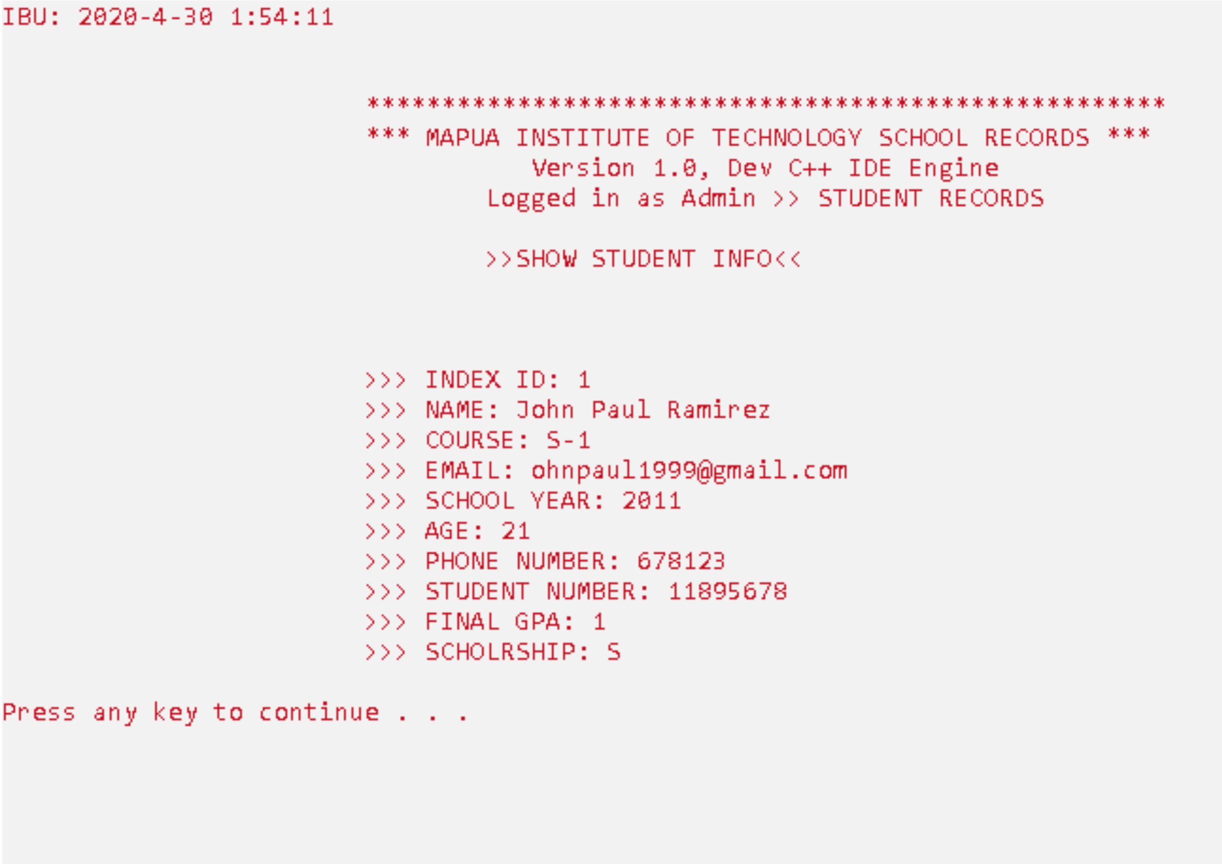
```
IBU: 2020-4-30 1:51:57


                        *****************************************************
                        *** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL RECORDS ***
                                   Version 1.0, Dev C++ IDE Engine
                                Logged in as Admin >> STUDENT RECORDS


                        [1]Upload Student Record

                        [2]Delete Student Record

                        [3]Update Student Record

                        [4]Search Student Record

                        [5]Show All Student Records

                        [6]GWA Calculator

                        [7]Export Records

                        [8]Go Back

                        Enter Choice:
```

This function enables the user to input the information of any given student. It is then recorded within the temporary cache
of the program (structure).



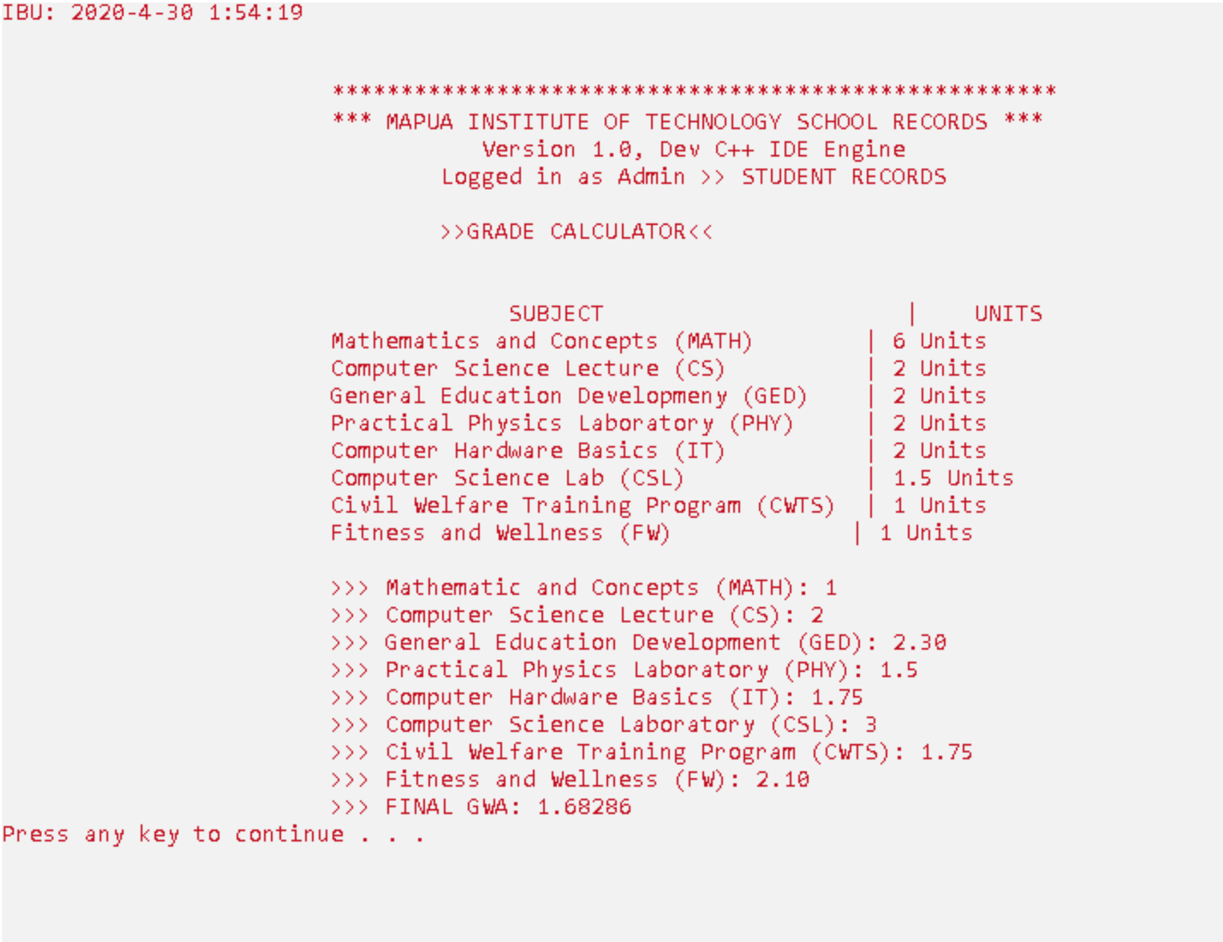
```
IBU: 2020-4-30 1:53:21


                        *****************************************************
                        *** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL RECORDS ***
                                   Version 1.0, Dev C++ IDE Engine
                                Logged in as Admin >> STUDENT RECORDS

                                >>UPLOAD STUDENT INFO<<


                        >>> Input Record Index Slot: 1
                        >>> Enter Full Name: John Paul Ramirez
                        >>> Enter Course: CS-1
                        >>> Enter Email: johnpaul1999@gmail.com
                        >>> Scholarship method (FS/HS/NONE): FS
                        >>> Enter Student Number: 2011895678
                        >>> Enter School Year Enrollment: 2011
                        >>> Enter Age: 21
                        >>> Enter Phone Number: 678123
                        >>> Enter Grade (GWA): 1.5
                        System Log: Updating. . .
                        System Log: Update has been completed!
Press any key to continue . . .
```

This function displays all the records that have been inputted by the user. It can hold up to 1000+ records (locally) for the purpose of this program.



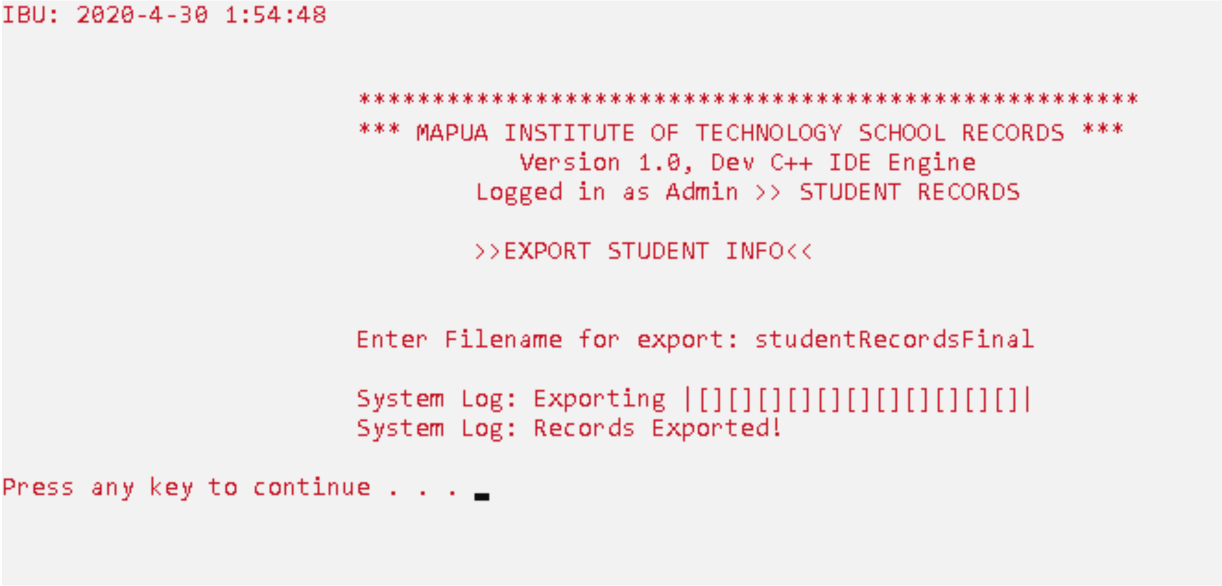
```
IBU: 2020-4-30 1:54:11


                        *****************************************************
                        *** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL RECORDS ***
                                   Version 1.0, Dev C++ IDE Engine
                                Logged in as Admin >> STUDENT RECORDS

                                >>SHOW STUDENT INFO<<



                        >>> INDEX ID: 1
                        >>> NAME: John Paul Ramirez
                        >>> COURSE: S-1
                        >>> EMAIL: ohnpaul1999@gmail.com
                        >>> SCHOOL YEAR: 2011
                        >>> AGE: 21
                        >>> PHONE NUMBER: 678123
                        >>> STUDENT NUMBER: 11895678
                        >>> FINAL GPA: 1
                        >>> SCHOLRSHIP: S

Press any key to continue . . .
```

This function enables the user to calculate a student’s GWA with a built-in calculator. The weighted average and units
may be modified within the code itself to suit the demands of the user.



```
IBU: 2020-4-30 1:54:19


                        *****************************************************
                        *** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL RECORDS ***
                                   Version 1.0, Dev C++ IDE Engine
                                Logged in as Admin >> STUDENT RECORDS

                                >>GRADE CALCULATOR<<


                                     SUBJECT                      |    UNITS
                        Mathematics and Concepts (MATH)        | 6 Units
                        Computer Science Lecture (CS)          | 2 Units
                        General Education Developmeny (GED)    | 2 Units
                        Practical Physics Laboratory (PHY)     | 2 Units
                        Computer Hardware Basics (IT)          | 2 Units
                        Computer Science Lab (CSL)             | 1.5 Units
                        Civil Welfare Training Program (CWTS)  | 1 Units
                        Fitness and Wellness (FW)             | 1 Units

                        >>> Mathematic and Concepts (MATH): 1
                        >>> Computer Science Lecture (CS): 2
                        >>> General Education Development (GED): 2.30
                        >>> Practical Physics Laboratory (PHY): 1.5
                        >>> Computer Hardware Basics (IT): 1.75
                        >>> Computer Science Laboratory (CSL): 3
                        >>> Civil Welfare Training Program (CWTS): 1.75
                        >>> Fitness and Wellness (FW): 2.10
                        >>> FINAL GWA: 1.68286
Press any key to continue . . .
```

This function outputs all records input by the user to an external file. The function also enables the user to give custom filenames to their records file. The time of export is also imprinted within the file itself.



```
IBU: 2020-4-30 1:54:48


                        ******************************************************
                        *** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL RECORDS ***
                                   Version 1.0, Dev C++ IDE Engine
                                Logged in as Admin >> STUDENT RECORDS

                                >>EXPORT STUDENT INFO<<


                        Enter Filename for export: studentRecordsFinal

                        System Log: Exporting |[][][][][][][][][][][]|
                        System Log: Records Exported!

Press any key to continue . . .
```

DOCUMENTATION

```
/*
        CS127-8L TERM PROJECT

        MAPUA STUDENT RECORDS

        Created By: Lorenzo Pastoral
        Date Created: March 26, 2020; 2:04pm

*/


/* Libraries */
#include<iostream>
#include<windows.h>
#include<cstdlib>
#include<conio.h>
#include<time.h>
#include<iomanip>
#include<cstring>
#include<fstream>
```

```
#include<ctime>
#include<cctype>
#include<cstdio>
#include<string>
#include<limits>
#include<ios>

using namespace std;

/* GLOBAL FUNCTIONS */
void getInitialize();
// Initializing sequence, no specific function
bool getLogin();
// Login sequence
void showTime();
// Displays current time
void mainMenu();
// Menu which will prompt the user to choose student or teacher
menu

/* STUDENT'S MENU */
void studentMenu();
    void uploadSRecord();
    void deleteSRecord();
    void searchSRecord();
    void updateSRecord();
    void showallSRecord();
    void calculateGrades();
```

```
        void displayGrades(struct studentGrades sGrades[], int size);
    void exportRecords();
    void exitProgram();

/* TEACHER'S MENU */
void teacherMenu();
    void uploadTRecord();
    void deleteTRecord();
    void searchTRecord();
    void updateTRecord();
    void showallTRecord();
    void exportTRecord();
    void exitProgram();

/* Global Variable Declaration For Accessing Data */
char error;
// Checks for error
bool inputFail;
// Checks if input of user corresponds to variable data type
string login_username;
// Declaration for username variable
string login_password;
// Declaration for password variable
string acctsFilename = "admininfo.dat";
// Declares filename of file streaming
int array = 0;
// Unique array name for student records
int array2 = 0;
```

```
// Unique array name for teacher records

/* Structures */
struct studentInfo {
    string name;
    string course;
    string email;
    string scholar;
    int schoolYear;
    int phoneNumber;
    int studentNumber;
    int age;
    int gwa;
    int recordInd;
};

struct teacherInfo {
    string name;
    string faculty;
    string email;
    string subject;
    int schoolYear;
    int teacherNumber;
    int phoneNumber;
    int age;
    int recordInd;
};
```

```
struct studentGrades {
    float gradeCS;
    float gradeCSL;
    float gradeIT;
    float gradeGED;
    float gradeFW;
    float gradeMATH;
    float gradePHY;
    float gradeCWTS;
    int recordInd;
};

/* SIZE OF STRUCTURE ARRAYS */
studentInfo sInfo[1000];
teacherInfo tInfo[1000];
studentGrades sGrades[1000];

/* Main Function */
int main()
{
    /* Declarations - Variables */
    bool validity;

    /* Background Color */
    system("color f4");

    /* Initialize Sequence */
    getInitialize();
    /* Check Validity of log-in */
    for(int ctr=0;;) {
        validity = getLogin();
            if(validity==true) {
                mainMenu();
            }
    }
    return 0;
}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void mainMenu()
{
    int mainChoice;

        /* Displays main menu */
        do{
        system("cls");
        showTime();
        cout << 
"\t\t\t****************************************************
*" << endl
            << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL 
RECORDS ***"<< endl
            << "\t\t\t\t\tVersion 1.0, Dev C++ IDE Engine             "<< 
endl
            << "\t\t\t\t\t\tLogged in as Admin                "<< endl << endl 
<< endl
            << "\t\t\t[1]Student Records" << endl << endl
            << "\t\t\t[2]Teacher Records" << endl << endl
            << "\t\t\t[3]Exit Program" << endl << endl
            << "\t\t\tEnter Choice: ";
        cin >> mainChoice;

            inputFail = cin.fail(); /* Checks if user failed to input an 
integer value */
            if(inputFail) {
                cout << "\t\t\tSystem Error: Invalid input! Please Try 
Again " << endl;
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
                system("pause");
            }
        }while(inputFail);

    switch(mainChoice)
    {
        case 1:
            studentMenu();
            break;
        case 2:
            teacherMenu();
            break;
        case 3:
            exit(0);
        default:
            cout << "\t\t\tSystem Error: Invalid input!  Please Try Again "
<< endl;
            cin.get(error);
            system("pause");
            mainMenu();
            break;
    }

}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
```

```
*********************/

void studentMenu()
{
    int studentChoice;

    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    /* Displays Student Menu */
    do{
        system("cls");
        showTime();
        cout <<
"\t\t\t****************************************************
*" << endl
            << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
            << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine              "<<
endl
            << "\t\t\t        Logged in as Admin >> STUDENT RECORDS
"<< endl << endl << endl
            << "\t\t\t[1]Upload Student Record" << endl << endl
            << "\t\t\t[2]Delete Student Record" << endl << endl
            << "\t\t\t[3]Update Student Record" << endl << endl
            << "\t\t\t[4]Search Student Record "<< endl << endl
            << "\t\t\t[5]Show All Student Records "<< endl << endl
            << "\t\t\t[6]GWA Calculator "<< endl << endl
            << "\t\t\t[7]Export Records"<< endl << endl
            << "\t\t\t[8]Go Back"<< endl << endl
            << "\t\t\tEnter Choice: ";
        cin >> studentChoice;

        inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try Again " << endl;
                cin.clear();
                cin.ignore(1000, '\n');
                system("pause");
            }

    }while(inputFail);



    switch(studentChoice)
    {
        case 1:
            uploadSRecord();
            break;
        case 2:
            deleteSRecord();
            break;
        case 3:
            updateSRecord();
            break;
        case 4:
            searchSRecord();
            break;
        case 5:
            showallSRecord();
            break;
        case 6:
            calculateGrades();
            break;
        case 7:
            exportRecords();
            break;
        case 8:
            mainMenu();
        default:
            cout << "\t\t\tSystem Error: Invalid input!  Please Try Again "
<< endl;
            cin.get(error);
            system("pause");
            studentMenu();
            break;


    }

}
```

```
/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void teacherMenu()
{

    int teacherChoice;

    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    /* Teacher Menu */
    system("cls");
    do{
        system("cls");
        showTime();
        cout <<
"\t\t\t****************************************************
*" << endl
            << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
```

```
RECORDS ***"<< endl
            << "\t\t\t\tVersion 1.0, Dev C++ IDE Engine             "<<
endl
            << "\t\t\t\t\tLogged in as Admin >> Teacher Records
"<< endl << endl << endl

            << "\t\t\t[1]Upload Teacher Record" << endl << endl
            << "\t\t\t[2]Delete Teacher Record" << endl << endl
            << "\t\t\t[3]Update Teacher Record" << endl << endl
            << "\t\t\t[4]Search Teacher Record "<< endl << endl
            << "\t\t\t[5]Show All Teacher Records "<< endl << endl
            << "\t\t\t[6]Export Records"<< endl << endl
            << "\t\t\t[7]Go Back"<< endl << endl
            << "\t\t\tEnter Choice: ";
        cin >> teacherChoice;
        inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }
        }while(inputFail);


    switch(teacherChoice)
    {
        case 1:
            uploadTRecord();
            break;
        case 2:
            deleteTRecord();
            break;
        case 3:
            updateTRecord();
            break;
        case 4:
            searchTRecord();
            break;
        case 5:
            showallTRecord();
            break;
        case 6:
            exportTRecord();
            break;
        case 7:
            mainMenu();
            break;
        default:
            cout<<"\t\t\t\t\t\tInvalid Input! \n\t\t\t\t\t\tPress any key to continue!";
            cin.get(error);
            system("pause");
            teacherMenu();
            break;

        }

    }

/
**********************************************************
*********************/
/
************************************DIVIDER***************
*********************/
/
**********************************************************
*********************/



                        // STUDENT CHOICES



/
**********************************************************
*********************/
/
************************************DIVIDER***************
*********************/
```

```
/
************************************************************
*********************/

void uploadSRecord()
{

        int idValue, idValue2;
        int idStart = 0;

    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    try {
        do{
        system("cls");
        showTime();
        cout <<
"\t\t\t********************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine             "<<
endl
             << "\t\t\t        Logged in as Admin >> STUDENT RECORDS
"<< endl << endl
             << "\t\t\t\t>>UPLOAD STUDENT INFO<<    "<< endl<<
```

```
endl << endl;
        cout << "\t\t\t>>> Input Record Index Slot: ";
        cin >> idValue;

            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
                system("pause");
            }
        }while(inputFail);

                /* Exception Block */
                if (idValue < 0 || idValue > 999) {
                    throw 99;
                }

    /* Inputs student information, loops through student structure
information */
    for(int j=0; j<=array; j++)
    {
        idValue2=idValue;

        if(idValue2==sInfo[j].recordInd)
        {
            idStart=1;
        }
    }
        if(idStart!=1)
        {
                sInfo[array].recordInd=idValue;

                do{
                    cout << "\t\t\t>>> Enter Full Name: ";
                    cin.ignore();
                    getline(cin,sInfo[array].name, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Course: ";
                    cin.ignore();
                    getline(cin,sInfo[array].course, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Email: ";
                    cin.ignore();
                    getline(cin,sInfo[array].email, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Scholarship method (FS/HS/NONE):
";
                    cin.ignore();
                    getline(cin,sInfo[array].scholar, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);


                do{
                    cout << "\t\t\t>>> Enter Student Number: ";
                    cin.ignore();
                    cin >> sInfo[array].studentNumber;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);
                do{
                    cout << "\t\t\t>>> Enter School Year Enrollment: ";
                    cin.ignore();
                    cin >> sInfo[array].schoolYear;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                cout << "\t\t\t>>> Enter Age: ";
                cin >> sInfo[array].age;
                    inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }
        }while(inputFail);
```

```
       do{
          cout << "\t\t\t>>> Enter Phone Number: ";
          cin.ignore();
          cin >> sInfo[array].phoneNumber;

          inputFail = cin.fail();
          if(inputFail)
          {
            cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
          }
        }while(inputFail);

        do{
          cout << "\t\t\t>>> Enter Grade (GWA): ";
          cin.ignore();
          cin >> sInfo[array].gwa;

          inputFail = cin.fail();
          if(inputFail)
          {
            cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
            cin.clear();
            cin.ignore(22, '\n');
          }
```

```
                }while(inputFail);

                array = array + 1;


                cout << "\t\t\tSystem Log: Updating";
                Sleep(0200);
                cout << ". ";
                Sleep(0200);
                cout << ". ";
                Sleep(0200);
                cout << ". " << endl;

                cout << "\t\t\tSystem Log: Update has been completed!"
<< endl;
                system("pause");
        }
        else {
            cout <<"\t\t\tSystem Error: This index has already been
entered." << endl;
            system("pause");
        }
    }

        catch(int) {
            cout << endl << "\t\t\tSystem Error: Error caught, index
value must be between 1-999 only!" << endl;
            system("pause");
            uploadSRecord();
        }


        catch(...) {
            cout << endl << "\t\t\tSystem Error: General error caught!"
<< endl;
            system("pause");
            uploadSRecord();
        }

            studentMenu();

}


/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/


void deleteSRecord()
```

```
{
    int idValue, idValue2;
    char confirmDelete;

    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    /* Input student slot to delete */
    try {
        do{
        system("cls");
        showTime();
        cout <<
"\t\t\t****************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine              "<<
endl
             << "\t\t\t        Logged in as Admin >> STUDENT RECORDS
"<< endl << endl
             << "\t\t\t\t>>DELETE STUDENT INFO<<    "<< endl<<
endl << endl

            << "\t\t\t>>> Enter Student ID Slot To Delete: ";
        cin >> idValue;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }
        }while(inputFail);

        /* Exception Block */
        if (idValue > 999 || idValue < 0) {
            throw 99;
        }

        /* Confirms deletion of student record */
        cout << "\t\t\t>>> CONFIRM DELETE? (THIS CANNOT BE UNDONE!) [Y/N]: ";
        cin >> confirmDelete;

        confirmDelete = toupper(confirmDelete);

        /* Checks if student accounte exists and confirms deletion */
        if(confirmDelete=='N')
        {
            deleteSRecord();
        }
        if(confirmDelete=='Y')
        {
            for(int j=0; j<= array; j++)
            {

            idValue2 = idValue;
                if(idValue2==sInfo[j].recordInd)
                {
                sInfo[j].recordInd='d';
                cout << "\t\t\t System Log: Student Record Deleted!";

                }
            }
        }
    }

    catch(int) {
            cout << endl << "\t\t\tSystem Error: Error caught, index
value must be between 1-999 only!" << endl;
            system("pause");
            deleteSRecord();
        }

    catch(...) {
            cout << endl << "\t\t\tSystem Error: General error caught!"
<< endl;
            system("pause");
            deleteSRecord();
        }
    cout << "\t\t\t";
    system("pause");
    studentMenu();

}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void updateSRecord()
{
    int idValue, idValue2;

    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    /* Input student slot to update */
    try {
        do{
        system("cls");
        showTime();
        cout <<
"\t\t\t*******************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine              "<<
endl
             << "\t\t\t        Logged in as Admin >> STUDENT RECORDS
"<< endl << endl
             << "\t\t\t\t>>UPDATE STUDENT INFO<<    "<< endl<<
endl << endl
             << "\t\t\t>>> Enter Student ID Slot to Update: ";
        cin >> idValue;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }
    }while(inputFail);

    /* Exception Block */
        if (idValue > 999 || idValue < 0) {
            throw 99;
        }
```

```
/* Iterate through J index records */
    for(int j=0; j<=array; j++)
    {

        idValue2 = idValue;
        if(idValue2==sInfo[j].recordInd)
        {
            cout << "\t\t\t***********************" << endl;
            cout << "\t\t\tStudent Index Number: ";
            cout << sInfo[j].recordInd << endl;

            cout << "\t\t\tFull Name: ";
            cout << sInfo[j].name << endl;

            cout << "\t\t\tCollege Course: ";
            cout << sInfo[j].course << endl;

            cout << "\t\t\tEmail: ";
            cout << sInfo[j].email << endl;

            cout << "\t\t\tYear Level: ";
            cout << sInfo[j].schoolYear << endl;

            cout << "\t\t\tStudent Number: ";
            cout << sInfo[j].studentNumber << endl;
```

```
                cout << "\t\t\tAge: ";
                cout << sInfo[j].age << endl;

                cout << "\t\t\tPhone Number: ";
                cout << sInfo[j].phoneNumber << endl;

                cout << "\t\t\tGWA: ";
                cout << sInfo[j].gwa << endl;

                cout << "\t\t\tScholarship: ";
                cout << sInfo[j].scholar << endl;
                cout << "\t\t\t***********************" << endl;

    cout << "\t\t\t***Re-enter Data***" << endl;

                do{
                    cout << "\t\t\t>>> Enter Full Name: ";
                    cin.ignore();
                    getline(cin,sInfo[j].name, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Course: ";
                    cin.ignore();
                    getline(cin,sInfo[j].course, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try 
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Email: ";
                    cin.ignore();
                    getline(cin,sInfo[j].email, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try 
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Scholarship Applicability: ";
                    cin.ignore();
                    getline(cin,sInfo[j].scholar, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);


                do{
                    cout << "\t\t\t>>> Enter Student Number: ";
                    cin.ignore();
                    cin >> sInfo[array].studentNumber;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter School Year: ";
                    cin.ignore();
                    cin >> sInfo[j].schoolYear;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                cout << "\t\t\t>>> Enter Age: ";
                cin >> sInfo[j].age;
                    inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try 
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }
        }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Phone Number: ";
                    cin.ignore();
                    cin >> sInfo[array].phoneNumber;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try 
Again " << endl;
                        cin.clear();
                        cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Grade (GWA): ";
                    cin.ignore();
                    cin >> sInfo[j].gwa;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                cout << "\t\t\tSystem Log: Updating";
                Sleep(0500);
                cout << ". ";
                Sleep(0500);
                cout << ". ";
                Sleep(0500);
                cout << ". " << endl;

                cout << "\t\t\tSystem Log: Update has been completed!"
<< endl;

            }
        }
    }

    catch(int) {
            cout << endl << "\t\t\tSystem Error: Error caught, index
value must be between 1-999 only!" << endl;
            system("pause");
            updateSRecord();
        }

    catch(...) {
            cout << endl << "\t\t\tSystem Error: General error caught!"
<< endl;
            system("pause");
            updateSRecord();
    }

        cout << "\t\t\t";
        system("pause");
        studentMenu();
}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void searchSRecord()
{
    int idValue, idValue2;
    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    /* Input student slot to search */
    try {
        do{
        system("cls");
        showTime();
        cout <<
"\t\t\t**************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine             "<<
endl
             << "\t\t\t        Logged in as Admin >> STUDENT RECORDS
"<< endl << endl
             << "\t\t\t\t>>SEARCH STUDENT INFO<<    "<< endl<<
endl << endl

            << "\t\t\t>>> Enter Student ID Slot Search: ";
        cin >> idValue;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
```

```
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }
        }while(inputFail);

        /* Exception Block */
        if (idValue > 999 || idValue < 0) {
            throw 99;
        }



        /* Iterate through J index loop */
        for(int j=0; j<=array; j++)
        {

            idValue2 = idValue;
            if(idValue2==sInfo[j].recordInd)
            {
                cout << "\t\t\tFull Name: ";
                cout << sInfo[j].name << endl;

                cout << "\t\t\tCollege Course: ";
                cout << sInfo[j].course << endl;

                cout << "\t\t\tYear Level: ";
                cout << sInfo[j].schoolYear << endl;
                cout << "\t\t\tStudent Number: ";
                cout << sInfo[j].studentNumber << endl;

                cout << "\t\t\tAge: ";
                cout << sInfo[j].age << endl;

                cout << "\t\t\tPhone Number: ";
                cout << sInfo[j].phoneNumber << endl;

                cout << "\t\t\tScholarship: ";
                cout << sInfo[j].scholar << endl;
            }
        }
    }

    catch(int) {
            cout << endl << "\t\t\tSystem Error: Error caught, index
value must be between 1-999 only!" << endl;
            system("pause");
            searchSRecord();
        }

    catch(...) {
            cout << endl << "\t\t\tSystem Error: General error caught!"
<< endl;
            system("pause");
            searchSRecord();
    }
```

```
        system("pause");
        studentMenu();
}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void showallSRecord()
{
    /* Displays all records in list form */
    system("cls");
        showTime();
        cout <<
"\t\t\t****************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine           "<<
endl
             << "\t\t\t        Logged in as Admin >> STUDENT RECORDS
```

```
"<< endl << endl
             << "\t\t\t\t>>SHOW STUDENT INFO<<  "<< endl<< endl
<< endl;

            for(int k=0; k<array; k++)
            {
                if(sInfo[k].recordInd!='d')
                {
                    cout << "\n";
                    cout << "\t\t\t>>> INDEX ID: " << sInfo[k].recordInd
<< endl;
                    cout << "\t\t\t>>> NAME: " << sInfo[k].name <<
endl;
                    cout << "\t\t\t>>> COURSE: " <<sInfo[k].course <<
endl;
                    cout << "\t\t\t>>> EMAIL: " <<sInfo[k].email <<
endl;
                    cout << "\t\t\t>>> SCHOOL YEAR: "
<<sInfo[k].schoolYear << endl;
                    cout << "\t\t\t>>> AGE: " <<sInfo[k].age << endl;
                    cout << "\t\t\t>>> PHONE NUMBER: "
<<sInfo[k].phoneNumber << endl;
                    cout << "\t\t\t>>> STUDENT NUMBER: "
<<sInfo[k].studentNumber << endl;
                    cout << "\t\t\t>>> FINAL GPA: " <<sInfo[k].gwa <<
endl;
                    cout << "\t\t\t>>> SCHOLRSHIP: " <<sInfo[k].scholar
<< endl;
```

```
                    }
            }
            cout << endl;

        system("pause");
        studentMenu();

}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void exportRecords()
{
    string fileName;
    char currDate[30];
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    sprintf(currDate, "%d-%d-%d %d:%d:%d", tm.tm_year + 1900,
tm.tm_mon + 1, tm.tm_mday, tm.tm_hour, tm.tm_min, tm.tm_sec);
```

```
    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

        system("cls");
        showTime();
        cout <<
"\t\t\t****************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine             "<<
endl
             << "\t\t\t        Logged in as Admin >> STUDENT RECORDS
"<< endl << endl
            << "\t\t\t\t\t>>EXPORT STUDENT INFO<< "<< endl<< endl
<< endl;

            /* Create file with custom filename*/
            cout << "\t\t\tEnter Filename for export: ";
            getline(cin, fileName);
            fileName += ".txt";

            /* Opens Student Records File*/
            ofstream studentRecords;
            studentRecords.open(fileName.c_str(), ios::app);
            /* Checks if file exists, if not, creates one*/
            if(!studentRecords.is_open())
            {
                cerr << "\t\t\tSystem Error: Cannot opening file!" <<
endl;
            }

            studentRecords <<
"\t\t\t****************************************************
*" << endl
                           << "\t\t\t*** MAPUA INSTITUTE OF
TECHNOLOGY SCHOOL RECORDS ***"<< endl
                           << "\t\t\t   OFFICIAL Rendered Student Records -
#10092356 - MSOITX"<< endl
                           << "\t\t\t     File Export Date: "<< currDate <<
endl << endl;



                for(int k = 0; k < array; k++)
                {
                    if(sInfo[k].recordInd!='d')
                    {
                    studentRecords << "\n";
                    studentRecords << "\t ";
                    studentRecords << "\t\t\t>>> INDEX ID: " <<
sInfo[k].recordInd << endl;
                    studentRecords << "\t\t\t>>> NAME: " <<
sInfo[k].name << endl;
                    studentRecords << "\t\t\t>>> COURSE: "
<<sInfo[k].course << endl;
                    studentRecords << "\t\t\t>>> EMAIL: "
<<sInfo[k].email << endl;
                    studentRecords << "\t\t\t>>> SCHOOL YEAR: "
<<sInfo[k].schoolYear << endl;
                    studentRecords << "\t\t\t>>> AGE: " <<sInfo[k].age
<< endl;
                    studentRecords << "\t\t\t>>> PHONE NUMBER: "
<<sInfo[k].phoneNumber << endl;
                    studentRecords << "\t\t\t>>> STUDENT NUMBER: "
<<sInfo[k].studentNumber << endl;
                    studentRecords << "\t\t\t>>> FINAL GPA: "
<<sInfo[k].gwa << endl;
                    studentRecords << "\t\t\t>>> SCHOLRSHIP: "
<<sInfo[k].scholar << endl;
                    }
                    studentRecords << endl;
                }

                cout << endl;

                cout << "\t\t\tSystem Log: Exporting |";
                Sleep(0600);
                cout << "[]";
                Sleep(0600);
                cout << "[]";
```

```
        Sleep(0600);
        cout << "[]";
        Sleep(0600);
        cout << "[]";
        Sleep(0600);
        cout << "[]";
        Sleep(0600);
        cout << "[]";
        Sleep(0600);
        cout << "[]";
        Sleep(0600);
        cout << "[]";
        Sleep(0600);
        cout << "[]";
        Sleep(0600);
        cout << "[]";
        Sleep(0600);
        cout << "[]|" << endl;

        cout << "\t\t\tSystem Log: Records Exported!" << endl
<< endl;

        studentRecords.close();

        system("pause");
        studentMenu();
}
```

```
/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************//
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void calculateGrades()
{
    int array = 0;

    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    system("cls");
    showTime();
            cout <<
"\t\t\t********************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine            "<<
endl
             << "\t\t\t        Logged in as Admin >> STUDENT RECORDS
"<< endl << endl
            << "\t\t\t\t>>GRADE CALCULATOR<<    "<< endl<< endl
<< endl;

        cout << "\t\t\t            SUBJECT                   |    UNITS " <<
endl
             << "\t\t\tMathematics and Concepts (MATH)        | 6
Units"<< endl
             << "\t\t\tComputer Science Lecture (CS)          | 2 Units"<<
endl
             << "\t\t\tGeneral Education Developmeny (GED)    | 2
Units"<< endl
             << "\t\t\tPractical Physics Laboratory (PHY)     | 2 Units"<<
endl
             << "\t\t\tComputer Hardware Basics (IT)          | 2 Units"<<
endl
             << "\t\t\tComputer Science Lab (CSL)             | 1.5 Units"<<
endl
             << "\t\t\tCivil Welfare Training Program (CWTS)  | 1
```

```
Units"<< endl
             << "\t\t\tFitness and Wellness (FW)            | 1 Units"<<
endl << endl;

        /* Input grades */
        do
        {
            cout <<"\t\t\t>>> Mathematic and Concepts (MATH): ";
            cin >> sGrades[array].gradeMATH;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }

            else if(sGrades[array].gradeMATH<1.00||
sGrades[array].gradeMATH>5.00)
            {
                cout << "System error: Must be between 1.00-5.00!" <<
endl;
                cin.clear();
            }
    }while(sGrades[array].gradeMATH<1.00||
sGrades[array].gradeMATH>5.00||inputFail);
```

```
        do
        {
            cout <<"\t\t\t>>> Computer Science Lecture (CS): ";
            cin >> sGrades[array].gradeCS;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }

            else if(sGrades[array].gradeCS<1.00||
sGrades[array].gradeCS>5.00)
            {
                cout << "System error: Must be between 1.00-5.00!" <<
endl;
                cin.clear();
            }
    }while(sGrades[array].gradeCS<1.00||
sGrades[array].gradeCS>5.00||inputFail);

        do
        {
            cout <<"\t\t\t>>> General Education Development (GED): ";
            cin >> sGrades[array].gradeGED;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << endl << "System Error: Invalid input! " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }

            else if(sGrades[array].gradeGED<1.00||
sGrades[array].gradeGED>5.00)
            {
                cout << "System error: Must be between 1.00-5.00!" <<
endl;
                cin.clear();
            }
    }while(sGrades[array].gradeGED<1.00||
sGrades[array].gradeGED>5.00||inputFail);


        do
        {
            cout <<"\t\t\t>>> Practical Physics Laboratory (PHY): ";
            cin >> sGrades[array].gradePHY;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }

            else if(sGrades[array].gradePHY<1.00||
sGrades[array].gradePHY>5.00)
            {
                cout << "System error: Must be between 1.00-5.00!" <<
endl;
                cin.clear();
            }
    }while(sGrades[array].gradePHY<1.00||
sGrades[array].gradePHY>5.00||inputFail);


        do
        {
            cout <<"\t\t\t>>> Computer Hardware Basics (IT): ";
            cin >> sGrades[array].gradeIT;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << endl << "System Error: Invalid input! " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }

            else if(sGrades[array].gradeIT<1.00||
sGrades[array].gradeIT>5.00)
```

```
            {
                cout << "System error: Must be between 1.00-5.00!" <<
endl;
                cin.clear();
            }
    }while(sGrades[array].gradeIT<1.00||
sGrades[array].gradeIT>5.00||inputFail);

        do
        {
            cout <<"\t\t\t>>> Computer Science Laboratory (CSL): ";
            cin >> sGrades[array].gradeCSL;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << endl << "System Error: Invalid input!" << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }

            else if(sGrades[array].gradeCSL<1.00||
sGrades[array].gradeCSL>5.00)
            {
                cout << "System error: Must be between 1.00-5.00!" <<
endl;
                cin.clear();
            }
    }while(sGrades[array].gradeCSL<1.00||
sGrades[array].gradeCSL>5.00||inputFail);


        do
        {
            cout <<"\t\t\t>>> Civil Welfare Training Program (CWTS): ";
            cin >> sGrades[array].gradeCWTS;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }

            else if(sGrades[array].gradeCWTS<1.00||
sGrades[array].gradeCWTS>5.00)
            {
                cout << "System error: Must be between 1.00-5.00!" <<
endl;
                cin.clear();
            }
    }while(sGrades[array].gradeCWTS<1.00||
sGrades[array].gradeCWTS>5.00||inputFail);

        do
        {
            cout <<"\t\t\t>>> Fitness and Wellness (FW): ";
            cin >> sGrades[array].gradeFW;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }

            else if(sGrades[array].gradeFW<1.00|| sGrades[array].gradeFW>5.00)
            {
                cout << "System error: Must be between 1.00-5.00!" << endl;
                cin.clear();
            }
    }while(sGrades[array].gradeFW<1.00|| sGrades[array].gradeFW>5.00||inputFail);

        /* Calls method to calculate grades and displays grades */
        displayGrades(sGrades, array);



}
```

```
void displayGrades(struct studentGrades sGrades[], int size)
{

    /* Weighs units and calculates GWA */
    float calcCS = sGrades[size].gradeCS * 2.00;
    float calcCSL = sGrades[size].gradeCSL * 1.50;
    float calcIT = sGrades[size].gradeIT * 2.00;
    float calcGED = sGrades[size].gradeGED * 2.00;
    float calcPHY = sGrades[size].gradePHY * 2.00;
    float calcMATH = sGrades[size].gradeMATH * 6.00;
    float calcCWTS = sGrades[size].gradeCWTS * 1.00;
    float calcFW = sGrades[size].gradeFW * 1.00;

    float finalGWA = ((calcCS + calcCSL + calcIT + calcGED +
calcMATH + calcPHY + calcCWTS + calcFW ) / 17.5);

    cout << "\t\t\t>>> FINAL GWA: " << finalGWA << endl;

    system("pause");
    studentMenu();
}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
```

```
/
*********************************************************
*********************/



                // TEACHER CHOICES



/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void uploadTRecord()
{

    int TidValue, TidValue2;
    int TidStart = 0;

  try {
    do{
        system("cls");
        showTime();
        cout <<
"\t\t\t*****************************************************
*" << endl
            << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
            << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine            "<<
endl
            << "\t\t\t        Logged in as Admin >> TEACHER RECORDS
"<< endl << endl
            << "\t\t\t\t>>UPLOAD TEACHER INFO<<    "<< endl<<
endl << endl;
        cout << "\t\t\t>>> Input Record Index Slot: ";
        cin >> TidValue;

                inputFail = cin.fail();
                if(inputFail)
                {
                    cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                    cin.clear();
                    cin.ignore(22, '\n');
                }
        }while(inputFail);

        /* Exception Block */
        if (TidValue < 0 || TidValue > 999) {
            throw 99;
        }



    for(int j2=0; j2<=array2; j2++) {
        TidValue2=TidValue;

        if(TidValue2==tInfo[j2].recordInd) {
            TidStart=1;
        }
    }
        if(TidStart!=1) {
                tInfo[array2].recordInd=TidValue;

                do{
                    cout << "\t\t\t>>> Enter Full Name: ";
                    cin.ignore();
                    getline(cin,tInfo[array2].name, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
        }while(inputFail);
```

```
                do{
                    cout << "\t\t\t>>> Enter Faculty Office: ";
                    cin.ignore();
                    getline(cin,tInfo[array2].faculty, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Email: ";
                    cin.ignore();
                    getline(cin,tInfo[array2].email, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Subject Code: ";
                    cin.ignore();
                    getline(cin,tInfo[array2].subject, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Teacher Number: ";
                    cin.ignore();
                    cin >>tInfo[array].teacherNumber;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
```

```
                        cin.clear();
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter School Year Enrollment: ";
                    cin.ignore();
                    cin >> tInfo[array2].schoolYear;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                cout << "\t\t\t>>> Enter Age: ";
                cin >> tInfo[array2].age;
                    inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }
        }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Phone Number: ";
                    cin.ignore();
                    cin >> tInfo[array].phoneNumber;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                array2 = array2 + 1;



                cout << "\t\t\tSystem Log: Updating";
                Sleep(0500);
                cout << ". ";
                Sleep(0500);
                cout << ". ";
                Sleep(0500);
                cout << ". " << endl;

                cout << "\t\t\tSystem Log: Update has been completed!"
<< endl;
                system("pause");
        }

        else
        {
            cout <<"\t\t\tSystem Error: This index has already been
entered." << endl;
            system("pause");
        }
    }

    catch(int) {
        cout << endl << "\t\t\tSystem Error: Error caught, index value
must be between 1-999 only!" << endl;
        system("pause");
        uploadTRecord();
    }

    catch(...) {
        cout << endl << "\t\t\tSystem Error: General error caught!" <<
endl;
        system("pause");
        uploadTRecord();
    }
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    teacherMenu();

}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void deleteTRecord()
{
    int TidValue, TidValue2;
    char confirmDelete;

    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    try {
        do {
```

```
        system("cls");
        showTime();
        cout <<
"\t\t\t**********************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine              "<<
endl
             << "\t\t\t        Logged in as Admin >> TEACHER RECORDS
"<< endl << endl
             << "\t\t\t\t>>DELETE TEACHER INFO<<    "<< endl<<
endl << endl

            << "\t\t\t>>> Enter Teacher ID Slot To Delete: ";
        cin >> TidValue;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }

        }while(inputFail);

        /* Exception Block */
        if (TidValue > 999 || TidValue < 0) {
            throw 99;
        }

        cout << "\t\t\t>>> CONFIRM DELETE? (THIS CANNOT BE
UNDONE!) [Y/N]: ";
        cin >> confirmDelete;

        confirmDelete = toupper(confirmDelete);

        if(confirmDelete=='N')
        {
            teacherMenu();
        }
        if(confirmDelete=='Y')
        {
            for(int j2=0; j2<= array2; j2++)
            {

            TidValue2 = TidValue;
                if(TidValue2==tInfo[j2].recordInd)
                {
                tInfo[j2].recordInd='d';
                cout << "\t\t\t System Log: Student Record Deleted!";

                }
            }
        }
```

```
    }

    catch(int) {
            cout << endl << "\t\t\tSystem Error: Error caught, index value must be between 1-999 only!" << endl;
            system("pause");
            deleteTRecord();
        }

    catch(...) {
        cout << endl << "\t\t\tSystem Error: General error caught!" << endl;
        system("pause");
        deleteTRecord();
    }

    cout << "\t\t\t";
    system("pause");
    teacherMenu();



}

/******************************************************************************/
/***********************************DIVIDER***************
```

```
*********************/
/
*********************************************************
*********************/

void updateTRecord()
{
    int TidValue, TidValue2;

    try {
        do{
        system("cls");
        showTime();
        cout <<
"\t\t\t****************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine            "<<
endl
             << "\t\t\t        Logged in as Admin >> TEACHER RECORDS
"<< endl << endl
             << "\t\t\t\t>>UPDATE TEACHER INFO<<    "<< endl<<
endl << endl



            << "\t\t\t>>> Enter Teacher ID Slot to Update: ";
        cin >> TidValue;
```

```
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }
    }while(inputFail);

        /* Exception Block */
        if (TidValue > 999 || TidValue < 0) {
            throw 99;
        }

        for(int j2=0; j2<=array2; j2++)
        {
            TidValue2 = TidValue;
            if(TidValue2==tInfo[j2].recordInd)
            {
                cout << "\t\t\t***********************" << endl;
                cout << "\t\t\tTeacher Index Number: ";
                cout << tInfo[j2].recordInd << endl;

                cout << "\t\t\tFull Name: ";
                cout << tInfo[j2].name << endl;

                cout << "\t\t\tFaculty Faculty: ";
                cout << tInfo[j2].faculty<< endl;

                cout << "\t\t\tSubject: ";
                cout << tInfo[j2].subject << endl;

                cout << "\t\t\tEmail: ";
                cout << tInfo[j2].email << endl;

                cout << "\t\t\tYear Level: ";
                cout << tInfo[j2].schoolYear << endl;

                cout << "\t\t\tTeacher Number: ";
                cout << tInfo[j2].teacherNumber << endl;

                cout << "\t\t\tAge: ";
                cout << tInfo[j2].age << endl;

                cout << "\t\t\tPhone Number: ";
                cout << tInfo[j2].phoneNumber << endl;

                cout << "\t\t\t***********************" << endl;
                cout << "\t\t\t***Re-enter Data***" << endl;

                do{
                cout << "\t\t\t>>> Enter Full Name: ";
                    cin.ignore();
                    getline(cin,tInfo[j2].name, '\n');
```

```
                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Faculty Office: ";
                    cin.ignore();
                    getline(cin,tInfo[j2].faculty, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Email: ";
                    cin.ignore();
                    getline(cin,tInfo[j2].email, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Subject Code: ";
                    cin.ignore();
                    getline(cin,tInfo[j2].subject, '\n');

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter Teacher Number: ";
                    cin.ignore();
                    cin >>tInfo[array].teacherNumber;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                    }
                }while(inputFail);

                do{
                    cout << "\t\t\t>>> Enter School Year: ";
                    cin.ignore();
                    cin >> tInfo[j2].schoolYear;

                    inputFail = cin.fail();
                    if(inputFail)
                    {
                        cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                        cin.clear();
                        cin.ignore(22, '\n');
                    }
                }while(inputFail);
```

```
do{
    cout << "\t\t\t>>> Enter Age: ";
    cin >> tInfo[j2].age;
    inputFail = cin.fail();
    if(inputFail)
    {
        cout << "\t\t\tSystem Error: Invalid input! Please Try Again " << endl;
        cin.clear();
        cin.ignore(5, '\n');
    }
}while(inputFail);

do{
    cout << "\t\t\t>>> Enter Phone Number: ";
    cin.ignore();
    cin >> tInfo[array].phoneNumber;

    inputFail = cin.fail();
    if(inputFail)
    {
        cout << "\t\t\tSystem Error: Invalid input! Please Try Again " << endl;
        cin.clear();
        cin.ignore(22, '\n');
    }
}while(inputFail);
```

```
                cout << "\t\t\tSystem Log: Updating";
                Sleep(0500);
                cout << ". ";
                Sleep(0500);
                cout << ". ";
                Sleep(0500);
                cout << ". " << endl;

                cout << "\t\t\tSystem Log: Update has been completed!" << endl;

            }
        }
    }

    catch(int) {
            cout << endl << "\t\t\tSystem Error: Error caught, index value must be between 1-999 only!" << endl;
            system("pause");
            updateTRecord();
        }

    catch(...) {
            cout << endl << "\t\t\tSystem Error: General error caught!" << endl;
            system("pause");
            updateTRecord();
        }

    cout << "\t\t\t";
    system("pause");
    teacherMenu();
}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void searchTRecord()
{
    int TidValue, TidValue2;

    try {
        do{
        system("cls");
        showTime();
        cout <<
"\t\t\t****************************************************
*" << endl
```

```
            << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
            << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine            "<<
endl
            << "\t\t\t        Logged in as Admin >> TEACHER RECORDS
"<< endl << endl
            << "\t\t\t\t>>SEARCH TEACHER INFO<<    "<< endl<<
endl << endl
            << "\t\t\t>>> Enter Teacher ID Slot Search: ";
        cin >> TidValue;
            inputFail = cin.fail();
            if(inputFail)
            {
                cout << "\t\t\tSystem Error: Invalid input! Please Try
Again " << endl;
                cin.clear();
                cin.ignore(5, '\n');
            }
        }while(inputFail);

        for(int j2=0; j2<=array2; j2++)
        {
            TidValue2 = TidValue;
            if(TidValue2==tInfo[j2].recordInd)
            {
            cout << "\t\t\t**********************" << endl;
                cout << "\t\t\tTeacher Index Number: ";
                cout << tInfo[j2].recordInd << endl;
                cout << "\t\t\tFull Name: ";
                cout << tInfo[j2].name << endl;

                cout << "\t\t\tFaculty Office: ";
                cout << tInfo[j2].faculty << endl;

                cout << "\t\t\tSubject: ";
                cout << tInfo[j2].subject << endl;

                cout << "\t\t\tEmail: ";
                cout << tInfo[j2].email << endl;

                cout << "\t\t\tYear Level: ";
                cout << tInfo[j2].schoolYear << endl;

                cout << "\t\t\tTeacher Number: ";
                cout << tInfo[j2].teacherNumber << endl;

                cout << "\t\t\tAge: ";
                cout << tInfo[j2].age << endl;

                cout << "\t\t\tPhone Number: ";
                cout << tInfo[j2].phoneNumber << endl;
                cout << "\t\t\t***********************" << endl;
            }
        }
    }
```

```
    catch(int) {
            cout << endl << "\t\t\tSystem Error: Error caught, index
value must be between 1-999 only!" << endl;
            system("pause");
            searchTRecord();
        }

    catch(...) {
            cout << endl << "\t\t\tSystem Error: General error caught!"
<< endl;
            system("pause");
            searchTRecord();
        }



    system("pause");
    teacherMenu();
}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
```

```
*********************/

void showallTRecord()
{

    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    system("cls");
        showTime();
        cout <<
"\t\t\t****************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine             "<<
endl
             << "\t\t\t        Logged in as Admin >> TEACHER RECORDS
"<< endl << endl
             << "\t\t\t\t>>SHOW TEACHER INFO<<  "<< endl<< endl
<< endl;

            for(int k2=0; k2<array2; k2++)
            {
                if(tInfo[k2].recordInd!='d')
                {
                cout << "\t\t\tTeacher Index Number: "
<<tInfo[k2].recordInd << endl;
                cout << "\t\t\tFull Name: " <<tInfo[k2].name << endl;
                cout << "\t\t\tFaculty Office: "<<tInfo[k2].faculty<< endl;
                cout << "\t\t\tSubject: " << tInfo[k2].subject << endl;
                cout << "\t\t\tEmail: " << tInfo[k2].email << endl;
                cout << "\t\t\tYear Level: " << tInfo[k2].schoolYear <<
endl    ;
                cout << "\t\t\tTeacher Number: " <<
tInfo[k2].teacherNumber << endl;
                cout << "\t\t\tAge: " << tInfo[k2].age << endl;
                cout << "\t\t\tPhone Number: " <<
tInfo[k2].phoneNumber << endl;
                    }
                cout << endl;
            }


        system("pause");
        teacherMenu();

}

/
************************************************************
*********************/
/
*********************************DIVIDER**************
*********************/
```

```
/
*********************************************************
*********************/

void exportTRecord()
{
    string fileName;
    char currDate[30];
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    sprintf(currDate, "%d-%d-%d %d:%d:%d", tm.tm_year + 1900,
tm.tm_mon + 1, tm.tm_mday, tm.tm_hour, tm.tm_min, tm.tm_sec);

    /* CLEAR BUFFER */
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

        system("cls");
        showTime();
        cout <<
"\t\t\t****************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\t   Version 1.0, Dev C++ IDE Engine            "<<
endl
             << "\t\t\t        Logged in as Admin >> TEACHER RECORDS
"<< endl << endl
```

```
            << "\t\t\t\t\t>>EXPORT TEACHER INFO<< "<< endl<< endl
<< endl;

            cout << "\t\t\tEnter Filename for export: ";
            getline(cin, fileName);
            fileName += ".txt";

            /* OPEN FILES */
            ofstream teacherRecords;
            teacherRecords.open(fileName.c_str(), ios::app);

            if(!teacherRecords.is_open())
            {
                cerr << "\t\t\tSystem log: Error opening file!" << endl;
            }



            teacherRecords <<
"\t\t\t********************************************************
*" << endl
                            << "\t\t\t*** MAPUA INSTITUTE OF
TECHNOLOGY SCHOOL RECORDS ***"<< endl
                            << "\t\t\t   OFFICIAL Rendered Teacher Records -
#10091356 - MSOITX"<< endl
                            << "\t\t\t    File Export Date: "<< currDate <<
endl << endl;

                for(int k2 = 0; k2 < array2; k2++)
                {
                    if(tInfo[k2].recordInd!='d')
                    {
                    teacherRecords << "\t\t\tTeacher Index Number: " << 
tInfo[k2].recordInd << endl;
                    teacherRecords << "\t\t\tFull Name: " << 
tInfo[k2].name << endl;
                    teacherRecords << "\t\t\tFaculty Office: "<< 
tInfo[k2].faculty << endl;
                    teacherRecords << "\t\t\tSubject: " << 
tInfo[k2].subject << endl;
                    teacherRecords << "\t\t\tEmail: " << tInfo[k2].email 
<< endl;
                    teacherRecords << "\t\t\tYear Level: " << 
tInfo[k2].schoolYear << endl;
                    teacherRecords << "\t\t\tTeacher Number: " << 
tInfo[k2].teacherNumber << endl;
                    teacherRecords << "\t\t\tAge: " << tInfo[k2].age << 
endl;
                    teacherRecords << "\t\t\tPhone Number: " << 
tInfo[k2].phoneNumber << endl;
                    }
                    teacherRecords << endl;
                }
                cout << endl;

                cout << "\t\t\tSystem Log: Exporting |";
                Sleep(0600);
```

```
          cout << "[]";
          Sleep(0600);
          cout << "[]";
          Sleep(0600);
          cout << "[]";
          Sleep(0600);
          cout << "[]";
          Sleep(0600);
          cout << "[]";
          Sleep(0600);
          cout << "[]";
          Sleep(0600);
          cout << "[]";
          Sleep(0600);
          cout << "[]";
          Sleep(0600);
          cout << "[]";
          Sleep(0600);
          cout << "[]";
          Sleep(0600);
          cout << "[]|" << endl;

          cout << "\t\t\tSystem Log: Records Exported!" << endl
<< endl;

          teacherRecords.close();

          system("pause");
```

```
            teacherMenu();

}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void getInitialize()
{

    cout << "Initializing Mapua School Records System (2019) --
Powered by Microsoft in Partenership With DOS Software (C++)";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "Booting up master records/C:user/pc/address/c++
project --- execute program this program!";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "Processing internal database // SSD 1 -- RAID 1
%temp%files%preconfig.atcm_datafiles%none%empty%cacheupto5
mp // ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "Accessing hardware provenance // Intel(R) Core(TM)
i3-5005U CPU @ 3.00GHZ - 2.00 GHz / 4.00 GB RAM / 64-bit
operating system";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "Automatic GPU acceleartion: true // Overide: Hardware
located (Nvidia GeForce 930M) -- Utilizes Direct3D: false";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "Creating vitrtual data space -- %origin%setup%
TTidValueLocation\\:3425sdf3s%df464sdfs57&";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "RAM MEMORY ALLOCATED -- 4.00 GB Available \\
Utilizing RAM by 100% -- 5464-4567687-456559834-657567-null";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "CPU Usage @ 5%";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "CPU Usage @ 15%";
    Sleep(0070);
```

```
cout << ". ";
Sleep(0070);
cout << ". ";
Sleep(0070);
cout << ". " << endl;
cout << "CPU Usage @ 29%";
Sleep(0070);
cout << ". ";
Sleep(0070);
cout << ". ";
Sleep(0070);
cout << ". " << endl;
cout << "CPU Usage @ 59%";
Sleep(0070);
cout << ". ";
Sleep(0070);
cout << ". ";
Sleep(0070);
cout << ". " << endl;
cout << "CPU Usage @ 74%";
Sleep(0070);
cout << ". ";
Sleep(0070);
cout << ". ";
Sleep(0070);
cout << ". " << endl;
cout << "CPU Usage @ 86%";
Sleep(0070);
```

```
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "CPU Usage @ 99%";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "Error log cache detcted: (50MB of de-allocated memory,
please declutter files! // x1009275684%x1093485&c://c++files/
cache)";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "Optimizing AVG Free Virus --------- 0 .12312% minimal
risks: no execeptions found!";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". ";
    Sleep(0070);
    cout << ". " << endl;
    cout << "Sending client info";
    Sleep(0500);
    cout << ". ";
    Sleep(0500);
    cout << ". ";
    Sleep(0500);
    cout << ". " << endl;


    cout << "\t\t\tSTART PROGRAM" << endl;
    Sleep(0700);

}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

void showTime()
{
    char currDate[30];




        time_t t = time(NULL);
        struct tm tm = *localtime(&t);
        sprintf(currDate, "%d-%d-%d %d:%d:%d", tm.tm_year +
1900, tm.tm_mon + 1, tm.tm_mday, tm.tm_hour, tm.tm_min,
tm.tm_sec);
        cout << "IBU: " << currDate << "\n\n\n";


}

/
*********************************************************
*********************/
/
***********************************DIVIDER***************
*********************/
/
*********************************************************
*********************/

bool getLogin()
{
    string username;
    char password[100], ch;
    ifstream inFile;
    ofstream outFile;

    system("cls");

    /* opens credtentials database which contains password and
username
    which can be found in root directory */

            inFile.open(acctsFilename.c_str());

            if(inFile.fail()) {

            cout << endl << "System Eror: CANNOT CREATE" <<
acctsFilename <<"!" << endl
            <<"System log: Creating a new dat file...." << endl;
Sleep(500);

            outFile.open(acctsFilename.c_str(), ios_base::app);
            outFile<<"admin\t\t\tadmin"<< endl;
            outFile.close();

            cout << "System log: dat file has been created..."<<endl;
Sleep(500);

            inFile.open(acctsFilename.c_str());
```

```
             }

        /* prompts login input */
        system("cls");
        cout <<
"\t\t\t****************************************************
*" << endl
             << "\t\t\t*** MAPUA INSTITUTE OF TECHNOLOGY SCHOOL
RECORDS ***"<< endl
             << "\t\t\t\tVersion 1.0, Dev C++ IDE Engine               "<<
endl << endl
             << "\t\t\t >>> Username: ";
        cin >> username;

        cout << "\t\t\t>>> Password: ";
        for(int ctr=0; ctr<100;) {
            ch = getch();
            if(ch==13) {
                password[ctr] == '\0';
                break;
            }
            else if (ch==8&&ctr>=1) {
                cout << "\b \b";
                ctr--;
                cout <<"*";
            }
            else if((ch>='a'&&ch<='z')||(ch>='A'&&ch<='Z')||
(ch>='0'&&ch<='9')) {
```

```
                password[ctr]=ch;
                ctr++;
                cout<<"*";
            }
        }

    /* checks validity of input by comparing it to the database file */
    do {
        inFile>>login_username>>login_password;
        if(login_username==username&&login_password==password) {
            return true;
        }
    }while(!inFile.eof());


        for(int ctr=0; ctr<20; ctr++) {
            password[ctr]='\0';
        }
        cout << endl << "\t\t\t System error: Invalid login!" << endl;
        system("pause");

        return false;



    inFile.close();

}
```