INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT018-3-1-ICP

INTRODUCTION TO C PROGRAMMING

APU-APD1F2209

HAND OUT DATE: 3 APRIL 2023

HAND IN DATE: 14 JUNE 2023

WEIGHTAGE: 50%

STUDENT NAME : DARSHRNNEY A/P PUSHPANATHAN

TP NUMBER : TP072253

INSTRUCTIONS TO CANDIDATES:

1. Submit your assignment online in Moodle unless advised otherwise
2. Late submission will be awarded zero(0) unless Extenuating Circumstances (EC) are upheld
3. Cases of plagiarism will be penalized
4. You must obtain at least 50% in each component to pass this module

Contents

INTRODUCTION

Programming a code for a café is an important job that someone must do. In APU I was given the opportunity to do this assignment and better help people to enrol themselves into a better coding session boot camp. Some of the assumptions I have made in this assignment is that, there would be 3 users which are , Admin , Tutor and Student . Admin will have full authority to add, update and display all the selected information that is necessary to be seen. Tutor on the other hand will have to log in their Tutor Code and password given by the administrator to check their classes, students enrolled in their class and their salary .Salary will be based on the amount of student they teach and amount of hours conducted (2hrs is RM100). Students will also have a similar option to log in their TP Number and passwords given by the administrator to check their classes their admin have registered or enrol themselves into additional classes. All default students and tutors are added into the system beforehand as well as additional students to display the various log in performances the system can do .

PSEUDOCODE

```
// Function to handle the admin section
function admin():
display "You are in Admin section"
    display available choices
    get choice_2 from user

     if choice_2 is 1:
        call add_tutor()
     else if choice_2 is 2:
        call update_tutor()
     else if choice_2 is 3:
        call display_tutor()
     else if choice_2 is 4:
        call add_session()
     else if choice_2 is 5:
        call update_session()
     else if choice_2 is 6:
        call display_session()
     else if choice_2 is 7:
        call add_stud()
     else if choice_2 is 8:
        call add_stud_session()

else if choice_2 is 9:
    call display_stud_session()
else if choice_2 is 10:
    display "Going back to the Menu!"
```

```
// Function to handle the tutor section
function tutor():
    display "You are in Tutor section"
    call t_log_in() function
    display available choices
    get choice_3 from user

    if choice_3 is 1:
        call display_tutor_session()
    else if choice_3 is 2:
        call display_tutor_stud_session()
    else if choice_3 is 3:
        call display_salary()
    else if choice_3 is 4:
        display "Going back to the Menu!"
```

```
// Function to handle the student section
function student():
    display "You are in Student section"
    call stud_log_in() function
    display available choices
    get choice_4 from user



if choice_4 is 1:
    call display_student_session()
else if choice_4 is 2:
    call enroll_student_session()
else if choice_4 is 3:
    display "Going back to the Menu!"
 if choice_4 is 1:
      call display_student_session()
    else if choice_4 is 2:
      call enroll_student_session()
    else if choice_4 is 3:
      display "Going back to the Menu!"


// Function to handle tutor login
function t_log_in():
   open Tutor.txt file
   get searchID from user

   while there are records in the file:
      read record from file

      if searchID matches the record's tutor code:
         get searchpass from user

         if searchpass matches the record's tutor password:
            display "SUCCESS!"
            close file
            return
```

```
// Function to handle tutor login
function t_log_in():
    open Tutor.txt file
    get searchID from user

    while there are records in the file:
        read record from file

        if searchID matches the record's tutor code:
            get searchpass from user

            if searchpass matches the record's tutor password:
                display "SUCCESS!"
                close file
                return

            else:
                display "Password incorrect"
                close file
                exit program

    display "The tutor record cannot be found!"
    close file
    exit program


// Function to handle student login
function stud_log_in():
    open Student.txt file
    get searchTP from user

    while there are records in the file:
        read record from file

        if searchTP matches the record's student TP:
            get search_pass from user

            if search_pass matches the record's student password:
                display "SUCCESS!"
                close file
                return

            else:
                display "Password incorrect"
                close file
                exit program

display "The Student record cannot be found!"
close file
exit program
```

```
// Function to add a tutor
function add_tutor():
    open Tutor.txt file
    get n from user

    for each tutor to save:
        get tutor details from user
        write tutor details to file

    close file
    display "Tutor record has been saved successfully"

// Function to update a tutor
function update_tutor():
    open Tutor.txt file
    open Tutor_tmp.txt file
    get searchID from user

    while there are records in the file:
        read record from file

        if searchID matches the record's tutor ID:
            get updated tutor details from user
            write updated tutor details to Tutor_tmp.txt file
            display "Tutor record has been updated successfully"

        else:
            write record to Tutor_tmp.txt file

    close Tutor.txt file
    close Tutor_tmp.txt file
    delete Tutor.txt file
    rename Tutor_tmp.txt to Tutor.txt

// Function to display tutor records
function display_tutor():
    open Tutor.txt file

    while there are records in the file:
        read record from file
        display record

    close file

// Function to add a session
function add_session():
    open Session.txt file
    get session details from user
    write session details to file
    close file
    display "Session has been added successfully"
```

```
// Function to update a session
function update_session():
    open Session.txt file
    open Session_tmp.txt file
    get searchID from user

    while there are records in the file:
        read record from file

        if searchID matches the record's session ID:
            get updated session details from user
            write updated session details to Session_tmp.txt file
            display "Session has been updated successfully"

                else:
                    write record to Session_tmp.txt file

            close Session.txt file
            close Session_tmp.txt file
            delete Session.txt file
            rename Session_tmp.txt to Session.txt
```

```
    // Function to display session records
    function display_session():
        open Session.txt file

        while there are records in the file:
            read record from file
            display record

        close file
//Function to add student
    function add_stud():
        open Student.txt file
        get stud details from user
        write stud details to file
        close file
        display "Student record has been saved successfully"

    // Function to enroll a student in a session
    function enroll_student_session():
        open Stud_Session.txt file
        get enrollment details from user
        write enrollment details to file
        close file
        display "Student has been enrolled successfully"
```

```
// Function to display student session records
function display_stud_session():
    open Stud_Session.txt file

    while there are records in the file:
        read record from file
        display record

    close file

// Function to display tutor sessions
function display_tutor_session():
    open Sessions.txt file
    get searchID from user

    // Array to store the session codes to search for
    sessionCodes = ["PYP101", "JAVA102", "CPL103", "WEB104", "CSP105"]
    numCodes = 5  // Number of session codes

    found = False

    for i in range(numCodes):
        if searchID is equal to sessionCodes[i]:
            found = True
            rewind file

        while there are records in the file:
            read record from file

            if searchID is equal to sess_code:
                display record

        break  // Exit the loop since the session code is found

if found is not equal to True:
    display "The Session record cannot be found!"

close file

// Function to display tutor-student session records
function display_tutor_stud_session():
    open Student_Session.txt file
    get searchID from user

    found = False

    while there are records in the file:
        read record from file

        if searchID is equal to sess_code:
            found = True
            display record
```

```
// Function to display tutor-student session records
function display_tutor_stud_session():
    open Student_Session.txt file
    get searchID from user

    found = False

    while there are records in the file:
        read record from file

        if searchID is equal to sess_code:
            found = True
            display record

if sess_code is equal to "JAV102" or sess_code is equal to "PYP101" or sess_code is equal to “CPL103”or sess_code is equal to “WEB104”, or sess_code is equal to “CSP105”:
                while there are records in the file:
                    read record from file

                    // Check if the sess_code is different, and if so, break the loop
                    if sess_code is not equal to searchID:
                        break
                    display record

                break // Exit the loop since the session code is found

        if found is not equal to True:
            display "The Session record cannot be found!"

        close file


// Function to display tutor salary for a session
function display_salary():
    open Student_Session.txt file
    get searchID from user

    // Array to store the session codes to search for
    sessionCodes = ["PYP101", "JAV102", "CPL103", "WEB104", "CSP105"]
    numCodes = 5  // Number of session codes

    found = False

    for i in range(numCodes):
        if searchID is equal to sessionCodes[i]:
            found = True
            rewind file
            lineCount = 0
```

```
// Function to display tutor salary for a session
function display_salary():
    open Student_Session.txt file
    get searchID from user

    // Array to store the session codes to search for
    sessionCodes = ["PYP101", "JAV102", "CPL103", "WEB104", "CSP105"]
    numCodes = 5  // Number of session codes

    found = False

    for i in range(numCodes):
        if searchID is equal to sessionCodes[i]:
            found = True
            rewind file
            lineCount = 0

    while there are records in the file:
        read record from file

        if searchID is equal to sess_code:
            lineCount += 1
            tot_hrs = End_time - Start_time
            salary = tot_hrs * lineCount
            display sess_code, tot_hrs, salary

            display "Total number of students for session code", searchID, ":", lineCount

            break  // Exit the loop since the session code is found

if found is not equal to True:
    display "The Session record cannot be found!"

close file


 // Function to display student session records
 function display_student_session():
    open Student_Session.txt file
    open Student_temp.txt file

    get searchID from user

    found = False

    while there are records in the file:
        read record from file

        if searchID is equal to stud_TP:
            found = True
            write record to Student_temp.txt file
```

```
    if found is equal to True:
        close Student_temp.txt file
        open Student_temp.txt file

        while there are records in the file:
while there are records in the file:
            read record from file
            display record

        close file
    else:
        display "The Student record cannot be found!"

    close files
    remove "Student_Check_session.txt" file
    rename "Student_temp.txt" file to "Student_Check_session.txt"

// Function to enroll student in a session
function enroll_student_session():
    open Student_Session.txt file
    open Student_temp.txt file

    get searchID from user

    found = False

    while there are records in the file:
        read record from file

        if searchID is equal to sess_code:
            found = True
            write record to Student_temp.txt file

    close files
if found is equal to True:
    open Student_temp.txt file

    lineCount = 0
    while there are records in the file:
        read record from file
        lineCount += 1

    n = 100 - lineCount
    display "There are", lineCount, "sessions already registered. There are", n, "spots available."

    get choice from user

    if choice is equal to "Y" or choice is equal to "y":
        open Student_Session.txt file in append mode
```

```
        get stud_TP from user
        get Fname from user

        write record to Student_Session.txt file
        display "Successfully added!"

        close file
    else:
        display "Session not added."

    close file
else:
    display "The Session record cannot be found!"



function main():
    choice = 0

    while True:
        display "HELLO!! WELCOME TO APU PROGRAMING CafE MANAGEMENT
SYSTEM**********************************************************************"
        display "Main Menu"
        display "1. Admin"
        display "2. Tutor"
        display "3. Student"
        display "4. Exit"
        display "Enter your choice: "
        input choice

if choice is equal to 1:
    admin()
else if choice is equal to 2:
    tutor()
else if choice is equal to 3:
    student()
else if choice is equal to 4:
    display "Exiting the program."
    return 0
else:
    display "Invalid choice. Please try again."
```

MAIN MENU

```c
//MAIN MENU !!
int main() {
    int choice;

    while (1) {
        // Display the main menu options
        printf("HELLO!! WELCOME TO APU PROGRAMING CafE MANAGEMENT SYSTEM**********************************************************
        printf("Main Menu\n");
        printf("1. Admin\n");
        printf("2. Tutor\n");
        printf("3. Student\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

         if (choice == 1) {
            admin();
        }
        else if (choice == 2) {
            tutor();
        }
        else if (choice == 3) {
```

In the main menu there are choices From : ADMIN , TUTOR ,STUDENT

Choices you can make later:

```c
void admin() {
    printf("You are in Admin section\n");

    int choice_2;
    printf("What would you like to do ?\n");
    printf("1.Add tutor \n");
    printf("2.Update tutor details\n");
    printf("3.Display Tutors\n");
    printf("4.Adding new programming café session\n");
    printf("5.Updating programming café session\n");
    printf("6.Display programming café session\n");
    printf("7.Registration of Student\n");
    printf("8.Enroll student in a session\n");
    printf("9.Listing of Programming Café sessions and participating students\n");
    printf("10.Exit\n");
    printf("Enter your choice :");
    scanf("%d", &choice_2);

    if(choice_2==1){
    add_tutor();
    }
    if(choice_2==2){
```

```c
void tutor() {
    printf("You are in Tutor section\n");
    t_log_in();
    int choice_3;
    printf("What would you like to do ?\n");
    printf("1.Display your classes? \n");
    printf("2.Display your students in your class?\n");
    printf("3.Display salary ?\n");
    printf("4.Exit\n");
    printf("Enter your choice :");
    scanf("%d", &choice_3);

    if(choice_3==1){
    display_tutor_session();
    }
    if(choice_3==2){
        display_tutor_stud_session();
    }
    if(choice_3==3){
        display_salary();
    }
    if(choice_3==4){
```

```c
void student() {
    printf("You are in Student section\n");
    stud_log_in();
    int choice_4;
    printf("What would you like to do ?\n");
    printf("1.Display your classes? \n");
    printf("2.Enroll in other avalaible classes?\n");
    printf("3.Exit\n");
    printf("Enter your choice :");
    scanf("%d", &choice_4);

    if(choice_4==1){
    display_student_session();
    }
    if(choice_4==2){
    enroll_student_session();
    }
    if(choice_4==3){
        printf("Going back to the Menu!\n");
        return 0;
}
```

DISPLAY

```
HELLO!! WELCOME TO APU PROGRAMING CafE MANAGEMENT SYSTEM**************************************************************
 Main Menu
1. Admin
2. Tutor
3. Student
4. Exit
Enter your choice:
```

LOG IN CODES

```c
void t_log_in() {
    FILE *old1;
    int t_pass, searchpass;
    char searchID[25], fname[25], lname[25], t_id[25], subject[25], t_code[25];
    old1 = fopen("Tutor.txt", "r");
    if (!old1) {
        printf("Error opening file...\n");
        exit(1);
    }

    int found = 0;
    printf("Enter the Tutor ID: ");
    scanf("%s", searchID);

    while (fscanf(old1, " %s %s %s %s %s %d", fname, lname, t_id, subject, t_code, &t_pass) != EOF) {
        if (strcmp(searchID, t_code) == 0) {
            found = 1;
            printf("The Tutor record is found.\nEnter the password for the tutor: ");
            scanf("%d", &searchpass);

    while (fscanf(old1, " %s %s %s %s %s %d", fname, lname, t_id, subject, t_code, &t_pass) != EOF){
            if (searchpass == t_pass) {
                printf("SUCCESS!!\n");
                fclose(old1);
                return;
            } else {
                printf("Password incorrect\n");
                fclose(old1);
```

```c
                printf("Password incorrect\n");
                fclose(old1);
                exit(1);
            }
        }
        }
    }

    if (found != 1) {
        printf("The tutor record cannot be found!\n");
        fclose(old1);
        exit(1);
    }

    fclose(old1);
    // Continue to the previous void
}
```

```c
void stud_log_in() {
    FILE *oldP;
    int stud_pass, search_pass;
    char searchTP[25], Fname[25], Lname[25], stud_TP[25];
    oldP = fopen("Student.txt", "r");
    if (!oldP) {
        printf("Error opening file...\n");
        exit(1);
    }

    int found = 0;
    printf("Enter the Student TP: ");
    scanf("%s", searchTP);

    while (fscanf(oldP, " %s | %s | %s | %d", Fname, Lname, stud_TP, &stud_pass) != EOF) {
        if (strcmp(searchTP, stud_TP) != 0) {
            found = 1;
            printf("The Student record is found.\nEnter the password for the student: ");
            scanf("%d", &search_pass);

            if (search_pass == stud_pass) {
                printf("SUCCESS!!\n");
                fclose(oldP);
                return;
            } else {
                printf("Password incorrect\n");
                fclose(oldP);
                exit(1);
            }
        }
    }

    if (found != 1) {
        printf("The Student record cannot be found!\n");
        fclose(oldP);
        exit(1);
    }

    fclose(oldP);
    // Continue to the previous void
}
```

DISPLAY

```
HELLO!! WELCOME TO APU PROGRAMING CafE MANAGEMENT SYSTEM=
 Main Menu
1. Admin
2. Tutor
3. Student
4. Exit
Enter your choice: 2
You are in Tutor section
Enter the Tutor ID: Albert123
The Tutor record is found.
Enter the password for the tutor: 1234
SUCCESS!!
What would you like to do ?
1.Display your classes?
```

```
Enter your choice: 3
You are in Student section
Enter the Student TP: TP1111
The Student record is found.
Enter the password for the student: 1234
SUCCESS!!
What would you like to do ?
1.Display your classes?
2.Enroll in other avalaible classes?
3.Exit
Enter your choice :
```

ADMIN CODES

```
//ADMIN SECTION*****************************************************************************************************************************
//TUTOR SECTION
//-------------------------------------------------------------------------------------------------------------------------------------
void add_tutor(){
    FILE *fptr;  //declare a file pointer
    char fname [25],lname[25],t_id[25],subject[25],t_code[25]; //Necessary variables
    int n,t_pass;
    int ans;
    fptr=fopen("Tutor.txt","a");
    if(fptr==NULL){
        printf("Error in opening file...\n");
        exit(1);
    }
    printf("Enter the number of Tutor to save : ");
    scanf("%d",&n);
    for(int i =0;i<n;i++){
        printf("Tutor No.%d ",i+1);
        printf("Enter tutor First Name :");
        scanf("%s",fname);
        printf("Enter tutor Last Name :");
        scanf("%s",lname);
        printf("Enter TutorID:");
        scanf("%s",&t_id);
        printf("Enter the subject:");
        scanf("%s",&subject);
        printf("Enter New tutor code for the proffessor :");
        scanf("%s",&t_code);
        printf("Enter New password for the proffessor (ONLY NUMBERS):");
```

```
        ans= fprintf(fptr,"%s | %s | %s | %s | %s %d\n",fname, lname,t_id,subject,t_code,t_pass);//How to store in a file

    }
    if (ans>0){
        printf("Tutor record has been saved successfully \n");

    }
    else{
        printf("unable to save the Tutor record\n");
    }
    fclose(fptr);
}
//-------------------------------------------------------------------------------------------------------------------------------------
```

```c
void update_tutor(){
    FILE *oldF,*newF;
    int t_pass;
    char searchID[25],fname[25],lname[25],t_id[25],subject[25],t_code[25];
    oldF=fopen("Tutor.txt","r");
    newF=fopen("Tutor_tmp.txt","w");
    if(!oldF|| !newF){
        printf("Error opening file..\n");
        exit(1);

    }
    int found=0;
    printf("Enter the Tutor ID to update :");
    scanf("%s",searchID);
    while(fscanf(oldF,"%s | %s | %s | %s | %s %d",fname,lname,t_id,subject,t_code,&t_pass)!=EOF){
    if (strcmp(searchID, t_id) == 0){
        found=1;
        printf("The Tutor record is found .Enter the new code for the tutor:\n");
        scanf("%s",t_code);
        printf("Enter new password for the tutor (ONLY NUMBERS):\n");
        scanf("%d",&t_pass);
        fprintf(newF,"%s | %s | %s | %s | %s %d\n",fname,lname,t_id,subject,t_code,t_pass);
        continue;//copying all the records
        }
        fprintf(newF,"%s | %s | %s | %s | %s %d\n",fname,lname,t_id,subject,t_code,t_pass);
        }
    }
    if (found!=1){
        printf("The tutor record cannot be found !\n");

    }

    fclose(oldF);
    fclose(newF);
    remove("Tutor.txt");
    rename("Tutor_tmp.txt","Tutor.txt");
}
//-----------------------------------------------------------------------------------------------------------------------
```

```c
void display_tutor() {
    FILE *fptr1;
    char read_tutor[80];
    fptr1 = fopen("Tutor.txt", "r");
    if (!fptr1) {
        printf("Error opening the file...\n");
        exit(1);
    }
    while (fgets(read_tutor, sizeof(read_tutor), fptr1) != NULL) {
        printf("%s |", read_tutor);
    }
    fclose(fptr1);
}
```

DISPLAY

```
Enter your choice: 1
You are in Admin section
What would you like to do ?
1.Add tutor
2.Update tutor details
3.Display Tutors
4.Adding new programming cafθ session
5.Updating programming cafθ session
6.Display programming cafθ session
7.Registration of Student
8.Enroll student in a session
9.Listing of Programming Cafθ sessions and participating students
10.Exit
Enter your choice :
```

```
Enter your choice :3
Albert | Eins | T01 | Web_dev | Albert123 1234
Ahmad | Ali | T02 | C_Sharp_Programming | Ahmad123 1234
Steve | Universe | T03 | Python_Programming | Steve123 1234
```

```
Enter your choice :6
JAV102 | JAVA_Programming | Sunday | 900 1100 2.00HR | C-01-02 | T02
CPL103 | C_Programming | Saturday | 1400 1600 2.00HR | C-01-03 | T03
WEB104 | Web_development | Sunday | 1400 1600 2.00HR | C-01-04 | T04
CSP105 | C_sharp_Programming | Monday | 1900 2100 2.00HR | C-01-05 | T05
PYP101 | Phython_Programming | Saturday | 900 1100 2.00HR | C-01-01 | T01
```

```
Enter your choice :8
Enter the available Session code: CSP105
The Session record is found.
How many students do you want to add to this session? :1
Enter the NO.1 Student name for the session: Abdullah
Enter Student TP for the session: TP1111
Enter Start time for the session (in 24hrs): 1900
Enter End time for the session (in 24hrs): 2100
Enter Location for the session: C-01-05
Enter Tutor code for the session: T05
```

```
Enter your choice :9
CPL103 | Mike | TP6969 | 1400 1600 | C-01-03 | T03
CPL103 | Fae | TP3333 | 1400 1600 | C-01-03 | T03
PYP101 | Yogeswaran | TP1212 | 900 1100 | C-01-01 | T01
CSP105 | Abdullah | TP1111 | 1900 2100 | C-01-05 | T05
```

```
271 void add_session(){
272     FILE *fptr2;  //declare a file pointer
273     char sess_code[25],title[25],day[25],Loc[25],T_code[25];//Necessary variables
274     int n,Start_time,End_time;
275     float tot_hrs;
276     int ans;
277     fptr2=fopen("Sessions.txt","a");
278     if(fptr2==NULL){
279         printf("Error in opening file...\n");
280         exit(1);
281     }
282     printf("Enter the new number of Sessions to add : ");
283     scanf("%d",&n);
284     for(int i =0;i<n;i++){
285         printf("Each session can only have one class per WEEK!\n");
286         printf("Session No.%d ",i+1);
287         printf("Enter Session Code :");
288         scanf("%s",sess_code);
289         printf("Enter Session name :");
290         scanf("%s",title);
291         printf("Enter day of the session:");
292         scanf("%s",day);
293         printf("Enter the Start Time of the session(in 24 hrs):");
294         scanf("%d",&Start_time);
295         printf("Enter End Time of the session (in 24 hrs):");
296         scanf("%d",&End_time);
297         tot_hrs=End_time-Start_time;
297         tot_hrs=End_time-Start_time;
298         tot_hrs=tot_hrs/100.00;
299         printf("Enter the location of the session:");
300         scanf("%s",Loc);
301         printf("Enter the lecturer ID, in charge of the session:");
302         scanf("%s",T_code);
303
304        ans= fprintf(fptr2,"%s | %s | %s | %d %d %.2fHR | %s | %s\n",sess_code,title,day,Start_time,End_time,tot_hrs,Loc,T_code);
305
306     }
307     if (ans>0){
308         printf("Session record has been saved successfully \n");
309
310     }
311     else{
312         printf("unable to save the Session record\n");
313     }
314     fclose(fptr2);
315 }
316 //-----------------------------------------------------------------------------------------------------------------
```

```c
id update_session(){
  FILE *oldS, *newS;
  int Start_time, End_time;
  char sess_code[25], title[25], day[25], Loc[25], T_code[25], searchID[25];
  float tot_hrs;
  oldS = fopen("Sessions.txt","r");
  newS = fopen("Sessions_tmp.txt","w");
  if(!oldS || !newS){
      printf("Error opening file..\n");
      exit(1);
  }
  int found = 0;
  printf("Enter the Session code to update: ");
  scanf("%s", searchID);
  while(fscanf(oldS,"%s | %s | %s | %d %d %.2fHR | %s | %s", sess_code, title, day, &Start_time, &End_time, &tot_hrs, Loc, T_code) != EOF){
      if(strcmp(searchID, sess_code) == 0){
          found = 1;
          printf("The Session record is found.\nEnter the new updated day for the session: ");
          scanf("%s", day);
          printf("Enter new Start time for the session (in 24hrs): ");
          scanf("%d", &Start_time);
          printf("Enter new End time for the session (in 24hrs): ");
          scanf("%d", &End_time);
          printf("Enter new Location for the session: ");
          scanf("%s", Loc);
          printf("Enter new Tutor code for the session: ");
          scanf("%s", T_code);
          tot_hrs = (float)(End_time - Start_time) / 100.0;
          fprintf(newS,"%s | %s | %s | %d %d %.2fHR | %s | %s\n", sess_code, title, day, Start_time, End_time, tot_hrs, Loc, T_code);
          continue;
      }
  }
  if(found != 1){
      printf("The Session record cannot be found!\n");
  }
  fclose(oldS);
  fclose(newS);
  remove("Sessions.txt");
  rename("Sessions_tmp.txt","Sessions.txt");
```

```c
id display_session() {
  FILE *fptr3;
  char read_session[80];
  fptr3 = fopen("Sessions.txt", "r");
  if (!fptr3) {
      printf("Error opening the file...\n");
      exit(1);
  }
  while (fgets(read_session, sizeof(read_session), fptr3) != NULL) {
      printf("%s", read_session);
  }
  fclose(fptr3);
```

DISPLAY

```
JAV102 | JAVA_Programming | Sunday | 900 1100 2.00HR | C-01-02 | T02
CPL103 | C_Programming | Saturday | 1400 1600 2.00HR | C-01-03 | T03
WEB104 | Web_development | Sunday | 1400 1600 2.00HR | C-01-04 | T04
CSP105 | C_sharp_Programming | Monday | 1900 2100 2.00HR | C-01-05 | T05
PYP101 | Phython_Programming | Saturday | 900 1100 2.00HR | C-01-01 | T01
```

```c
STUDENT SECTION
-----------------------------------------------------------------------------------------------------------------------
id add_stud() {
  FILE *fptr4;
  char Fname[25], Lname[25], stud_TP[25];
  int n, stud_pass;
  int ans;

  fptr4 = fopen("Student.txt", "a");
  if (fptr4 == NULL) {
      printf("Error in opening file...\n");
      exit(1);
  }

  printf("Enter the number of students to save: ");
  scanf("%d", &n);

  for (int i = 0; i < n; i++) {
      printf("Student No.%d\n", i + 1);
      printf("Enter Student First Name: ");
      scanf("%s", Fname);
      printf("Enter Student Last Name: ");
      scanf("%s", Lname);
      printf("Enter Student TP: ");
      scanf("%s", stud_TP);
      printf("Enter New password for Student (ONLY NUMBERS): ");
      scanf("%d", &stud_pass);

      ans = fprintf(fptr4, "%d | %s | %s | %s | %d\n", i + 1, Fname, Lname, stud_TP, stud_pass);
  }
```

```c
  if (ans > 0) {
      printf("Student records have been saved successfully.\n");
  } else {
      printf("Unable to save the student records.\n");
  }

  fclose(fptr4);

-----------------------------------------------------------------------------------------------------------------------
```

```c
id add_stud_session() {
  FILE *oldSS, *newSS, *tempSS;
  int Start_time, End_time, n;
  char sess_code[25], title[25], day[25], Loc[25], T_code[25], searchID[25], Fname[25], stud_TP[25];
  float tot_hrs;
  char line[100];
  oldSS = fopen("Sessions.txt", "r");
  newSS = fopen("Student_Session.txt", "r");
  tempSS = fopen("Student_Session_tmp.txt", "w");
  if (!oldSS || !newSS || !tempSS) {
      printf("Error opening file...\n");
      exit(1);
  }
  int found = 0;
  printf("Enter the available Session code: ");
  scanf("%s", searchID);
  while (fscanf(oldSS, "%s | %s | %s | %d %d %.2fHR | %s | %s", sess_code, title, day, &Start_time, &End_time, &tot_hrs, Loc, T_code) != EOF) {
      if (strcmp(searchID, sess_code) == 0) {
          found = 1;
          break;
      }
  }
  if (found) {
      printf("The Session record is found.\n");
      printf("How many students do you want to add in this session? :");
      scanf("%d", &n);
      for (int i = 0; i < n; i++) {
          printf("Enter the NO.%d Student name for the session: ", i + 1);
          scanf("%s", Fname);
          printf("Enter Student ID for the session: ");
          scanf("%s", stud_TP);
          printf("Enter Start time for the session (in 24hrs): ");
          scanf("%d", &Start_time);
          printf("Enter End time for the session (in 24hrs): ");
          scanf("%d", &End_time);
          printf("Enter Location for the session: ");
          scanf("%s", Loc);
          printf("Enter Tutor code for the session: ");
          scanf("%s", T_code);
          fprintf(tempSS, "%s | %s | %s | %d %d | %s | %s\n", sess_code, Fname, stud_TP, Start_time, End_time, Loc, T_code);
      }
  } else {
      printf("The Session record cannot be found!\n");
  }
  // Copy the existing contents from "Student_Session.txt" to the temporary file
  while (fgets(line, sizeof(line), newSS) != NULL) {
      fprintf(tempSS, "%s", line);
  }
  printf("Successfully added !");
  fclose(oldSS);
  fclose(newSS);
  fclose(tempSS);
  remove("Student_Session.txt");
  rename("Student_Session_tmp.txt", "Student_Session.txt");
```

```c
id display_stud_session() {
  FILE *fptr5;
  char read_stud_sess[500];
  fptr5 = fopen("Student_session.txt", "r");
  if (!fptr5) {
      printf("Error opening the file...\n");
      exit(1);
  }
  while (fgets(read_stud_sess, sizeof(read_stud_sess), fptr5) != NULL) {
      printf("%s", read_stud_sess);
  }
  fclose(fptr5);
```

```
Enter the number of students to save: 3
Student No.1
Enter Student First Name: Aymen
Enter Student Last Name: Ansari
Enter Student TP: TP6969
Enter New password for Student (ONLY NUMBERS): 1234
Student No.2
Enter Student First Name: Lubaina
Enter Student Last Name: Khan
Enter Student TP: TP2222
Enter New password for Student (ONLY NUMBERS): 1234
Student No.3
Enter Student First Name: Zack
```

```
Enter your choice :8
Enter the available Session code: JAV102
The Session record is found.
How many students do you want to add in this session? :2
Enter the NO.1 Student name for the session: Zack
Enter Student ID for the session: TP4343
Enter Start time for the session (in 24hrs): 0900
Enter End time for the session (in 24hrs): 1100
Enter Location for the session: C-01-02
Enter Tutor code for the session: T02
Enter the NO.2 Student name for the session: Lubaina
Enter Student ID for the session: TP2222
```

```
Enter your choice :9
JAV102 | Zack | TP4343 | 900 1100 | C-01-02 | T02
JAV102 | Lubaina | TP2222 | 900 1100 | C-01-02 | T02
CPL103 | Eliza | TP5555 | 1400 1600 | C-01-03 | T03
PYP101 | Rick | TP1010 | 900 1100 | C-01-01 | T01
PYP101 | Gary | TP9999 | 900 1100 | C-01-01 | T01
PYP101 | Tae | TP0001 | 900 1100 | C-01-01 | T01
PYP101 | John | TP1111 | 900 1100 | C-01-01 | T01
```

TUTOR

```c
//TUTOR SECTION*************************************************************************************************************************
//-------------------------------------------------------------------------------------------------------------------------------------
void display_tutor_session() {
    FILE *fptr6;
    float tot_hrs;
    int Start_time, End_time;
    char sess_code[25], title[25], day[25], Loc[50], T_code[50], searchID[25], Fname[25], stud_TP[25];
    fptr6 = fopen("Sessions.txt", "r");
    if (!fptr6) {
        printf("Error opening the file...\n");
        exit(1);
    }

    // Array to store the session codes to search for
    char sessionCodes[5][25] = {"PYP101", "JAVA102", "CPL103", "WEB104", "CSP105"};
    int numCodes = 5;  // Number of session codes

    printf("Enter the session code: ");
    scanf("%s", searchID);

    int found = 0;
    for (int i = 0; i < numCodes; i++) {
        if (strcmp(searchID, sessionCodes[i]) == 0) {
            found = 1;
            rewind(fptr6);
            while (fscanf(fptr6, "%s | %s | %s | %d %d %fHR | %s ", sess_code, title, day, &Start_time, &End_time, &tot_hrs, Loc) != EOF) {
                if (strcmp(searchID, sess_code) == 0) {
                    printf("%s | %s | %s | %d %d %fHR | %s \n", sess_code, title, day, Start_time, End_time, tot_hrs, Loc);
                }
            }
            break;  // Exit the loop since the session code is found
        }
    }

    if (found != 1) {
        printf("The Session record cannot be found!\n");
    }

    fclose(fptr6);
}

//-------------------------------------------------------------------------------------------------------------------------------------
```

```c
void display_tutor_stud_session() {
    FILE *fptr7;
    int Start_time, End_time;
    char sess_code[25], title[25], day[25], Loc[50], T_code[50], searchID[25], Fname[25], stud_TP[25], read_stud_sess[100];
    fptr7 = fopen("Student_session.txt", "r");
    if (!fptr7) {
        printf("Error opening the file...\n");
        exit(1);
    }

    int found = 0;
    printf("Enter the available Session code: ");
    scanf("%s", searchID);

    while (fscanf(fptr7, "%s | %s | %s | %d %d | %s | %s", sess_code, Fname, stud_TP, &Start_time, &End_time, Loc, T_code) != EOF) {
        if (strcmp(searchID, sess_code) == 0) {
            found = 1;
            printf("%s | %s | %s | %d %d | %s | %s\n", sess_code, Fname, stud_TP, Start_time, End_time, Loc, T_code);

            // If the selected sess_code is JAV102, PYP101, or CPL103, print additional lines
            if (strcmp(sess_code, "JAV102") == 0 || strcmp(sess_code, "PYP101") == 0 || strcmp(sess_code, "CPL103") == 0) {
                while (fscanf(fptr7, "%s | %s | %s | %d %d | %s | %s", sess_code, Fname, stud_TP, &Start_time, &End_time, Loc, T_code) != EOF) {
                    // Check if the sess_code is different, and if so, break the loop
                    if (strcmp(sess_code, searchID) != 0) {
                        break;
                    }
                    printf("%s | %s | %s | %d %d | %s | %s\n", sess_code, Fname, stud_TP, Start_time, End_time, Loc, T_code);
                }
            }

            break;  // Exit the loop since the session code is found
        }
    }

    if (found != 1) {
        printf("The Session record cannot be found!\n");
    }

    fclose(fptr7);
}
```

```c
//-----------------------------------------------------------------------------------------------------------------
void display_salary() {
    FILE *fptr8;
    float salary, tot_hrs;
    int Start_time, End_time;
    char sess_code[25], title[25], day[25], Loc[50], T_code[50], searchID[25], Fname[25], stud_TP[25];

    fptr8 = fopen("Student_session.txt", "r");
    if (!fptr8) {
        printf("Error opening the file...\n");
        exit(1);
    }

    // Array to store the session codes to search for
    char sessionCodes[5][25] = {"PYP101", "JAV102", "CPL103", "WEB104", "CSP105"};
    int numCodes = 5;  // Number of session codes

    printf("Enter the session code: ");
    scanf("%s", searchID);

    int found = 0;
    for (int i = 0; i < numCodes; i++) {
        if (strcmp(searchID, sessionCodes[i]) == 0) {
            found = 1;
            rewind(fptr8);

            int lineCount = 0;
            while (fscanf(fptr8, "%s | %s | %s | %d %d %fHR | %s\n", sess_code, title, day, &Start_time, &End_time, &tot_hrs, Loc) != EOF) {
                if (strcmp(searchID, sess_code) == 0) {
                    lineCount++;
                    tot_hrs = End_time - Start_time;
                    salary = tot_hrs * lineCount;
                    printf("%s | %.2f | RM%.2f\n", sess_code, tot_hrs, salary);
                }
            }

            printf("Total number of students for session code %s: %d\n", searchID, lineCount);
            break;  // Exit the loop since the session code is found
        }
    }

    if (found != 1) {
        printf("The Session record cannot be found!\n");
    }

    fclose(fptr8);
}
```

```
Enter your choice: 2
You are in Tutor section
What would you like to do ?
1.Display your classes?
2.Display your students in your class?
3.Display salary ?
4.Exit
Enter your choice :
```

```
Enter your choice :1
Enter the session code: PYP101
PYP101 | Phython_Programming | Saturday | 900 1100 2.000000HR | C-01-01
```

```
Enter your choice :2
Enter the available Session code: JAV102
JAV102 | Zack | TP4343 | 900 1100 | C-01-02 | T02
JAV102 | Lubaina | TP2222 | 900 1100 | C-01-02 | T02
```

```
Enter your choice :2
Enter the available Session code: PYP101
PYP101 | Yogeswaran | TP1212 | 900 1100 | C-01-01 | T01
```

```
Enter your choice :3
Enter the session code: PYP101
PYP101 | 200.00 | RM200.00
Total number of students for session code PYP101: 1
```

STUDENT

```c
//STUDENT ********************************************************************************************************************************
//--------------------------------------------------------------------------------------------------------------------------------------
void display_student_session() {
    FILE *fptr9, *newFF;
    float tot_hrs;
    int Start_time, End_time;
    char sess_code[25], title[25], day[25], Loc[50], T_code[50], searchID[25], Fname[25], stud_TP[25], read_stud_sess[100];
    fptr9 = fopen("Student_Session.txt", "r");
    newFF = fopen("Student_temp.txt", "w");
    if (!fptr9 || !newFF) {
        printf("Error opening the file...\n");
        exit(1);
    }

    int found = 0;
    printf("Enter the TP Number: ");
    scanf("%s", searchID);

    while (fscanf(fptr9, "%s | %s | %s | %d %d | %s | %s", sess_code, Fname, stud_TP, &Start_time, &End_time, Loc, T_code) != EOF) {
        if (strcmp(searchID, stud_TP) == 0) {
            found = 1;
            fprintf(newFF, "%s | %s | %s | %d %d | %s | %s\n", sess_code, Fname, stud_TP, Start_time, End_time, Loc, T_code);
        }
    }

    if (found == 1) {
        fclose(newFF);
        newFF = fopen("Student_temp.txt", "r");
        if (!newFF) {
            printf("Error opening the file...\n");
            exit(1);
        }

        while (fgets(read_stud_sess, sizeof(read_stud_sess), newFF) != NULL) {
            printf("%s", read_stud_sess);
        }
        fclose(newFF);}
        else {
        printf("The Session record cannot be found!\n");
        }

    fclose(fptr9);
    remove("Student_Check_session.txt");
    rename("Student_temp.txt", "Student_Check_session.txt");
}
//--------------------------------------------------------------------------------------------------------------------------------------
```

```c
void enroll_student_session() {
    FILE *fptr0, *newTT, *studentSessionFile;
    int n, Start_time, End_time;
    char read_stud_sess[500];
    char searchID[25], choice[25];
    char sess_code[25], Fname[25], stud_TP[25], day[25], Loc[50], T_code[50], title[25];

    fptr0 = fopen("Student_session.txt", "r");
    newTT = fopen("Student_temp.txt", "w");

    if (!fptr0 || !newTT) {
        printf("Error opening the file...\n");
        exit(1);
    }

    printf("Enter the available Session code: ");
    scanf("%s", searchID);

    int found = 0;
    while (fscanf(fptr0, "%s | %s | %s | %d %d | %s | %s", sess_code, Fname, stud_TP, &Start_time, &End_time, Loc, T_code) != EOF) {
        if (strcmp(searchID, sess_code) == 0) {
            found = 1;
            fprintf(newTT, "%s | %s | %s | %d %d | %s | %s\n", sess_code, Fname, stud_TP, Start_time, End_time, Loc, T_code);
        }
    }

    fclose(fptr0);
    fclose(newTT);
```

```c
    if (found == 1) {
        newTT = fopen("Student_temp.txt", "r");
        if (!newTT) {
            printf("Error opening the file...\n");
            exit(1);
        }

        int lineCount = 0;
        while (fgets(read_stud_sess, sizeof(read_stud_sess), newTT) != NULL) {
            lineCount++;
        }

        n = 100 - lineCount;
        printf("There are %d sessions already registered. There are %d spots available.\n", lineCount, n);

        printf("Press (Y/N) to add this session into your timetable: ");
        scanf("%s", choice);

        if (choice[0] == 'Y' || choice[0] == 'y') {
            studentSessionFile = fopen("Student_session.txt", "a");  // Open in append mode
            if (!studentSessionFile) {
                printf("Error opening the file...\n");
                exit(1);
            }
            printf("Enter your TP Number: ");
            scanf("%s", stud_TP);
            printf("Enter your name: ");
            scanf("%s", Fname);
```

```
            printf("Enter your name: ");
            scanf("%s", Fname);

            fprintf(studentSessionFile, "%s | %s | %s | %d %d | %s | %s\n", sess_code, Fname, stud_TP, Start_time, End_time, Loc, T_code);
            printf("Successfully added!\n");
            fclose(studentSessionFile);
        } else {
            printf("Session not added.\n");
        }

        fclose(newTT);
    } else {
        printf("The Session record cannot be found!\n");
    }
}
```

```
Enter your choice: 3
You are in Student section
What would you like to do ?
1.Display your classes?
2.Enroll in other avalaible classes?
3.Exit
Enter your choice :1
Enter the TP Number: TP4343
JAV102 | Zack | TP4343 | 900 1100 | C-01-02 | T02
```

```
Enter your choice :2
Enter the available Session code: PYP101
There are 6 sessions already registered. There are 94 spots available.
Press (Y/N) to add this session into your timetable: y
Enter your TP Number: TP4343
Enter your name: Zack
Successfully added!
```

After :

```
You are in Student section
What would you like to do ?
1.Display your classes?
2.Enroll in other avalaible classes?
3.Exit
Enter your choice :1
Enter the TP Number: TP4343
JAV102 | Zack | TP4343 | 900 1100 | C-01-02 | T02
PYP101 | Zack | TP4343 | 900 1100 | C-01-01 | T01
```

CONCLUSION

In conclusion I have what I have learned in this assignment is the key usage of file pointers . Not only did I use them in various different forms , but I have leaened to use this technique in any given situation to retrieve a value from a text file. I have also learned that if I have determination to finish this code , I am able to succeed in creating a code with the help of my teacher . He taught us key skills such as control structures, functions, array, pointers, structures, unions and files in our program throughout this module. I am able to apply these skills and practice comments , variables ,naming conventions and indentations as well. I have also learned how to write a pseudocode for the given code I have programmed .

From this APU Programming Café Management system , I have many various outlooks on where we can use this similar program into .Programs such as this can be used in corporate settings to identify the different roles each individual plays , or we can use program like this in in a Educational institution or Professional settings .

Future advancements that we can take a look at is , we can further develop the log in information for it to have user authentication and authorization . We could also use all the information better by coding it and giving us a report and analysis on the raw data. This will provide us with valuable information that would help them make the right decisions on how to treat each individuals properly. We could integrate a calendar system which would benefit the admins in making a schedule .This would prevent them from clashing with other time and organize their work flow in a timely manner.

REFERENCE

C arrays. Tutorials. (n.d.). https://www.w3schools.com/c/c_arrays.php#:~:text=Arrays%20are%20used%20to%20store,separate%20variables%20for%20each%20value.

C pointers. Programiz. (n.d.). https://www.programiz.com/c-programming/c-pointers

GeeksforGeeks. (2021, June 22). *How to append a character to a string in C*. GeeksforGeeks. https://www.geeksforgeeks.org/how-to-append-a-character-to-a-string-in-c/

GeeksforGeeks. (2023, April 26). *Difference between structure and union in C*. GeeksforGeeks. https://www.geeksforgeeks.org/structure-vs-union-in-c/

How coding is impacting the future of Technology. Nonprofit Coding and Cybersecurity Bootcamp. (n.d.). https://www.elevenfifty.org/blog/how-coding-is-impacting-the-future-of-technology

Sneh. (2022, August 3). *Fgets() and gets() in C Programming*. DigitalOcean. https://www.digitalocean.com/community/tutorials/fgets-and-gets-in-c-programming

Strcmp() in C C. strcmp() in C C. (n.d.). https://www.tutorialspoint.com/strcmp-in-c-cplusplus#:~:text=strcmp()%20in%20C%2FC%2B%2B&text=The%20function%20strcmp()%20is,the%20strings%20character%20by%20character.