Logo, company name

Description automatically generated

**TECHNOLOGY PARK MALAYSIA**

Shape

**INTRODUCTION TO C PROGRAMMING**

**INDIVIDUAL ASSIGNMENT**

**NAME : MOHAMMAD ABU HUZAIFA**

**TP NUMBER : TP070586**

**MODULE CODE : CT018-3-1-ICP**

**INTAKE CODE : APD1F2209CS**

**LECTURER NAME : ASSOC. PROF. DR. NG KENG HOONG**

**HAND OUT WEEK : 4TH/5TH WEEK OF SEMESTER**

**SUBMISSION DATE : 14th JUNE 2023**

**MAIL :** [ng.kenghoong@apu.edu.my](mailto:ng.kenghoong@apu.edu.my)

I declare that:

* We understand what is meant by plagiarism.
* The implication of plagiarism have been explained to us by our lecturer.
* This project is all our work and we have acknowledged any use of the published or unpublished works of other people.

# Abstract

Programming Café is a plenary, conducted to provide an additional coding session for students outside their regular timetable concerning improving the student’s performance, and enhancing their problem-solving skills and confidence level in coding, organized by APU (Asia Pacific University of Technology and Innovation). APU has selected several popular programming languages for programming cafes during this pilot period called programming Python, Java, Web Development, C, and C Sharp. This program considers the use of control structures, functions, arrays, pointers, structs, associations, and files. The program must apply the module programming techniques and must be menu driven. In addition, this session on different segments will be displayed on the selected agenda. The aim of this project is to write a program in C language for the APU to manage Programming Cafe Sessions based on several categories. The system must be able to manage by three specialists in their respective functional areas. The relevant features associated with the user type will be provided automatically in the system based on the credentials.

Table of Contents

[Abstract 2](#_Toc137677646)

[1.0 Introduction & Assumptions 4](#_Toc137677647)

[1.1 Introduction 4](#_Toc137677648)

[1.2 Assumptions 4](#_Toc137677649)

[2.0 Design of the Program (Pseudocode) 5](#_Toc137677650)

[2.1 Main Menu Section 5](#_Toc137677651)

[2.2 Admin Login Section 6](#_Toc137677652)

[2.3 Admin Menu Section 7](#_Toc137677653)

[2.3.1 Add Tutor Section 9](#_Toc137677654)

[2.3.2 Add Student Section 10](#_Toc137677655)

[2.3.3 Add Programming Cafe Session Section 11](#_Toc137677656)

[2.3.4 Display Tutor Record Section 12](#_Toc137677657)

[2.3.5 Display Programming Cafe Session Section 13](#_Toc137677658)

[2.3.6 Display Student Record Section 14](#_Toc137677659)

[2.3.7 Search Record by Tutor ID Section 14](#_Toc137677660)

[2.3.8 Search Programming Cafe Session by Session Code Section 16](#_Toc137677661)

[2.3.9 Search Student by Student ID Section 17](#_Toc137677662)

[2.3.10 Modify Tutor Record Section 18](#_Toc137677663)

[2.3.11 Modify Student Record Section 21](#_Toc137677664)

[2.3.12 Modify Programming Cafe Session Record Section 23](#_Toc137677665)

[2.3.13 Enroll Student in a Session Section 25](#_Toc137677666)

[2.4 Tutor Login Section 27](#_Toc137677667)

[2.5 Tutor Menu Section 28](#_Toc137677668)

[2.5.1 Search Session timing by Tutor ID Section 29](#_Toc137677669)

[2.5.2 Search Participating Student List Section by Tutor ID 30](#_Toc137677670)

[2.6 Student Login Section 32](#_Toc137677671)

[2.7 Student Menu Section 34](#_Toc137677672)

[2.7.1 Display Student Session Section 34](#_Toc137677673)

[3.0 Additional Features 36](#_Toc137677674)

[4.0 Sample Outputs 41](#_Toc137677675)

[5.0 Conclusion 48](#_Toc137677676)

[6.0 References 48](#_Toc137677677)

# 1.0 Introduction & Assumptions

## Introduction

Welcome to the APU Programming Café Management System. As part of this project, APU (Asia Pacific University) will provide a C programming solution to manage the Programming Café Sessions. The Programming Café is a pilot program created to provide extra coding lessons for students to improve their problem-solving abilities and coding confidence.

An all-encompassing system designed for three user categories (Admin, Tutor, Student) is needed for this task. Features and capabilities unique to each user type have been included into the system.

The system should allow the admin user to do things like register tutors, create new programming cade sessions, register students, enroll students in sessions, and display a list of sessions and the students that are enrolled in them.

On the other side, Tutor needs to be able to see the sessions that have been allocated to them in addition to accessing the session listings.

Students have the capability to observe the sessions that have been allocated to them, as well as the choice to register themselves in any session that is open for enrollment.

Proper validations and logical assumptions should be established and recorded to assure the correctness and dependability of system. Good programming practices, including the use of comments, consistent naming conventions for variables, and adequate indentation, along with the judicious use of symbolic constants, are expected.

Additionally, it is recommended that you do exhaustive testing of the system using both positive and negative test scenarios. Input/output images and detailed descriptions of test results should be included in the assignment document to prove that the proposed solution really works.

## Assumptions

As per the requirement of the given task, I assume that -

1. The System is able to provide three user types [Admin , Tutor, Student]
2. The System will require certain credentials to allow access to dedicated sections
3. The Admin staff will be able to add and records of :

* Tutor
* Student
* Programming Café Session

1. The Admin Staff will be able to Display records of :

* Tutor
* Student
* Programming Café Session

1. The Admin staff will be able to search for specific records of :

* Tutor by Tutor ID
* Student by Student ID (TP Number)
* Programming Cafe Session by Session ID

1. The Admin staff will able to modify the records of :

* Tutor
* Student
* Programming Cafe Session

1. The Admin Staff will able to

* Enrol student in a session

1. Registered Tutors will be able to :

* Search & Display sessions timing by Tutor ID
* Search & Display participating student details by Tutor ID

1. Registered students will be able to :

* Search & Display Session timing by TP Number

# 2.0 Design of the Program (Pseudocode)

## 2.1 Main Menu Section

Function mainmenu()

Display “WELCOME TO APU PROGRAMMING CAFE MANAGEMENT SYSTEM”

Display “Choose User Type (1/2/3)”

Display “1. ADMIN”

Display “2. TUTOR”

Display “3. STUDENT”

Display “0. Terminate Program”

Input menuc1

If menuc1 == 1 then

Display “You’ve Chosen The Admin User Type”

Call adminlogin()

Else if menuc1 == 2 then

Display “You’ve Chosen The Tutor User Type”

Call tutorlogin()

Else if menuc1 == 3 then

Display “You’ve Chosen The Student User Type”

Call studentlogin()

Else

Display “Wrong Input”

Call mainmenu()

End if

End function

Function main()

Call mainmenu()

Return 0

End function

## 2.2 Admin Login Section

function adminlogin()

display "LOGIN TO USE ADMIN SYSTEMS"

input adminpass

if adminpass == 444602 then

display "Login Successful"

call adminmenu()

else

display "Wrong Password!"

7call adminlogin()

end if

end function

## 2.3 Admin Menu Section

function adminlogin()

display "LOGIN TO USE ADMIN SYSTEMS"

input adminpass

if adminpass == 444602 then

display "Login Successful"

call adminmenu()

else

display "Wrong Password!"

call adminlogin()

end if

end function

function adminmenu()

display "WELCOME TO APU PROGRAMMING CAFE MANAGEMENT ADMIN SYSTEMS"

display "Here Are The Functionalities!"

display "1. Add Records of"

display " 11. Tutor"

display " 12. Student"

display " 13. Programming Cafe Session"

display "2. Display All Records of"

display " 21. Tutor"

display " 22. Programming Cafe Session"

display " 23. Registered Students"

display "3. Search Specific Records of"

display " 31. Tutor by Tutor ID"

display " 32. Programming Cafe Session by Session ID"

display " 33. Student by Student ID"

display "4. Modify Record of"

display " 41. Tutor"

display " 42. Student"

display " 43. Programming Cafe Session"

display "5. Enroll"

display " 51. Student in a Session"

display "0. Exit"

input adminc1

switch adminc1

case 11:

call addtutor()

case 12:

call addstudent()

case 13:

call addprgcafses()

case 21:

call distutor()

case 22:

call disprgcafses()

case 23:

call disregstud()

case 31:

call distutorid()

case 32:

call dissessionid()

case 33:

call disstudid()

case 41:

call modtutor()

case 42:

call modstudent()

case 43:

call modprgcafses()

case 51:

call addstudses()

case 0:

call mainmenu()

default:

display "Invalid choice"

call adminmenu()

end switch

end function

### 2.3.1 Add Tutor Section

function addtutor()

display "Tutor Records"

open file "Tutor Records.txt" in append mode and assign it to tutorrec

display "Enter Tutor Details"

declare variables Tutorid, Pass, Name, DOB, Prgses, Prgsescode as strings

display "Tutor ID: "

input Tutorid

display "Password: "

input Pass

display "Name: "

input Name

display "Date of Birth: "

input DOB

display "Programming Session: "

input Prgses

display "Programming Session Code: "

input Prgsescode

write Tutorid, Pass, Name, DOB, Prgses, Prgsescode to tutorrec

close tutorrec

declare variable X as integer

display "Do You Want To Continue? [0 / 1]: "

input X

if X == 0 then

call adminmenu()

else if X == 1 then

call addtutor()

else

display "Wrong Input"

call mainmenu()

end if

end function

### 2.3.2 Add Student Section

function addstudent()

display "Student Records"

open file "Student Records.txt" in append mode and assign it to studentrec

if studentrec is null then

display "Failed to open Student Records.txt"

return

end if

display "Enter Student Details"

declare variables Name, DOB, Tpnumber, Pass as strings

display "Enter Student Name: "

input Name

display "Enter Date of Birth: "

input DOB

display "Enter TP Number: "

input Tpnumber

display "Enter Password: "

input Pass

write Name, DOB, Tpnumber, Pass to studentrec

close studentrec

declare variable X as integer

display "Do You Want To Continue? [0 / 1]: "

input X

if X == 0 then

call adminmenu()

else if X == 1 then

call addstudent()

else

display "Wrong Input"

call mainmenu()

end if

end function

### 2.3.3 Add Programming Cafe Session Section

function addprgcafses()

display "Programming Cafe Session Record"

open file "Programming Cafe Session.txt" in append mode and assign it to prgsesrec

if prgsesrec is null then

display "Failed to open Programming Cafe Session.txt"

return

end if

display "Enter Programming Cafe Details"

declare variables prgsescode, Name, Wkday, Time, Location, Tutor\_ID as strings

display "Enter Programming Session Code: "

input prgsescode

display "Enter Programming Session Name: "

input Name

display "Enter Day of The Week: "

input Wkday

display "Enter Time of The Day [HH:MM AM/PM]: "

input Time

display "Enter Session Location: "

input Location

display "Enter Tutor ID for this Session: "

input Tutor\_ID

write prgsescode, Name, Wkday, Time, Location, Tutor\_ID to prgsesrec

close prgsesrec

declare variable X as integer

display "Do You Want To Continue? [0 / 1]: "

input X

if X == 0 then

call adminmenu()

else if X == 1 then

call addprgcafses()

else

display "Wrong Input"

call mainmenu()

end if

end function

### 2.3.4 Display Tutor Record Section

function distutor()

display "Tutor\_ID\tPassword\tName\tDate\_of\_Birth\tProgramming\_Session\tSession\_Code"

open file "Tutor Records.txt" in read mode and assign it to tutorrec

if tutorrec is null then

display "Failed to open Tutor Records.txt"

return

end if

declare variable line as string

while read a line from tutorrec and store it in line

display line

end while

close tutorrec

declare variable X as integer

display "Go Back To Admin? [0 / 1]: "

input X

if X == 0 then

call adminmenu()

else if X == 1 then

display "Good Bye!"

else

display "Wrong Input"

call mainmenu()

end if

end function

### 2.3.5 Display Programming Cafe Session Section

function disprgcafses()

display "Session\_code\tSession\_name\t\tWKday\t\tTime\tLocation\tTutor\_ID"

open file "Programming Cafe Session.txt" in read mode and assign it to prgsesrec

if prgsesrec is null then

display "Failed to open Programming Cafe Session.txt"

return

end if

declare variable line as string

while read a line from prgsesrec and store it in line

display line

end while

close prgsesrec

declare variable X as integer

display "Go Back To Admin? [0 / 1]: "

input X

if X == 0 then

call adminmenu()

else if X == 1 then

display "Good Bye!"

end if

end function

### 2.3.6 Display Student Record Section

function disregstud()

open file "Student Records.txt" in read mode and assign it to studentrec

if studentrec is null then

display "Error opening file."

return

end if

declare variable c as character

display "Name\t\tDate\_of\_Birth\tTP\_Number\tPass"

while read a character from studentrec and assign it to c

display c

end while

close studentrec

declare variable X as integer

display "Go Back To Admin? [0 / 1] : "

input X

if X == 0 then

call adminmenu()

else if X == 1 then

display "Good Bye!"

end if

end function

### 2.3.7 Search Record by Tutor ID Section

function distutorid()

open file "Tutor Records.txt" in read mode and assign it to tutorrec

if tutorrec is null then

display "Error opening file."

return

end if

declare variable sid as string

declare variable str as string

declare variable flag as integer

declare variable pos as character

display "Enter Tutor ID To Search : "

input sid

set flag to 0

while read a line from tutorrec and assign it to str

set pos to the first occurrence of sid in str

if pos is not null then

display "Tutor\_ID\tPassword\tName\tDate\_of\_Birth\tProgramming\_Session\tSession\_code"

display str

set flag to flag + 1

break the loop

end if

end while

if flag is equal to 0 then

display "No Entry Found For Tutor ID : '<sid>' !!!"

end if

close tutorrec

declare variable X as integer

display "Go Back To Admin? [0 / 1] : "

input X

if X is equal to 0 then

call adminmenu()

else if X is equal to 1 then

display "Good Bye!"

end if

end function

### 2.3.8 Search Programming Cafe Session by Session Code Section

function dissessionid()

open file "Programming Cafe Session.txt" in read mode and assign it to prgsesrec

if prgsesrec is null then

display "Error opening file."

return

end if

declare variable spid as string

declare variable str as string

declare variable flag as integer

declare variable pos as character

display "Enter Session Code to Search: "

input spid

set flag to 0

while read a line from prgsesrec and assign it to str

set pos to the first occurrence of spid in str

if pos is not null then

display "Session\_code\tSession\_name\t\tWKday\t\tTime\tLocation\tTutor\_ID"

display str

set flag to flag + 1

break the loop

end if

end while

if flag is equal to 0 then

display "No Entry Found For Session Code : '<spid>' !!!"

end if

declare variable X as integer

display "Go Back To Admin? [0 / 1] : "

input X

if X is equal to 0 then

call adminmenu()

else if X is equal to 1 then

display "Good Bye!"

end if

end function

### 2.3.9 Search Student by Student ID Section

function disstudid()

open file "Student Records.txt" in read mode and assign it to studentrec

if studentrec is null then

display "Error opening file."

return

end if

declare variable stid as string

declare variable str as string

declare variable flag as integer

declare variable pos as character

display "Enter TP Number to Search : "

input stid

set flag to 0

while read a line from studentrec and assign it to str

set pos to the first occurrence of stid in str

if pos is not null then

display "Name\t\tDate\_of\_Birth\tTP\_Number\tPass"

display str

set flag to flag + 1

break the loop

end if

end while

if flag is equal to 0 then

display "No Entry Found For TP Number : '<stid>' !!!"

end if

declare variable X as integer

display "Go Back To Admin? [0 / 1] : "

input X

if X is equal to 0 then

call adminmenu()

else if X is equal to 1 then

display "Good Bye!"

end if

end function

### 2.3.10 Modify Tutor Record Section

function modtutor()

display "Tutor\_ID\tPassword\tName\tDate\_of\_Birth\tProgramming\_Session\tSession\_code"

open file "Tutor Records.txt" in read mode and assign it to tutorrec

if tutorrec is null then

display "Error opening file."

return

end if

display newline

declare variable line as string

while read a line from tutorrec and assign it to line

display line

end while

close tutorrec

declare variables O and Y as strings

display "Enter The Word : "

input O

remove newline character from O

display "Replacement : "

input Y

remove newline character from Y

open file "out.txt" in write mode and assign it to temp

open file "Tutor Records.txt" in read mode and assign it to tutorrec

if temp or tutorrec is null then

display "Error opening file."

return

end if

while read a line from tutorrec and assign it to line

write line to temp

end while

close tutorrec

close temp

open file "out.txt" in read mode and assign it to temp

open file "Tutor Records.txt" in write mode and assign it to tutorrec

if temp or tutorrec is null then

display "Error opening file."

return

end if

while read a line from temp and assign it to line

declare variable pos as string

while pos is found by searching for O in line

set index to the position of pos in line

copy Y to pos

copy line starting from index + length(O) to the end to pos + length(Y)

set the character at line[length(line) - length(O) + length(Y)] to null

end while

write line to tutorrec

end while

close temp

close tutorrec

declare variable X as integer

display "Go Back To Admin? [0 / 1] : "

input X

if X is equal to 0 then

call adminmenu()

else if X is equal to 1 then

display "Good Bye!"

end if

end function

### 2.3.11 Modify Student Record Section

function modstudent()

display "Name\t\tDate\_of\_Birth\tTP\_Number\tPass"

open file "Student Records.txt" in read mode and assign it to studentrec

if studentrec is null then

display "Error opening file."

return

end if

display newline

declare variable line as string

while read a line from studentrec and assign it to line

display line

end while

close studentrec

declare variables O and Y as strings

display "Choose The Entry : "

input O

remove newline character from O

display "Replacement : "

input Y

remove newline character from Y

open file "out.txt" in write mode and assign it to temp

open file "Student Records.txt" in read mode and assign it to studentrec

if temp or studentrec is null then

display "Error opening file."

return

end if

while read a line from studentrec and assign it to line

write line to temp

end while

close studentrec

close temp

open file "out.txt" in read mode and assign it to temp

open file "Student Records.txt" in write mode and assign it to studentrec

if temp or studentrec is null then

display "Error opening file."

return

end if

while read a line from temp and assign it to line

declare variable pos as string

while pos is found by searching for O in line

set index to the position of pos in line

copy Y to pos

copy line starting from index + length(O) to the end to pos + length(Y)

set the character at line[length(line) - length(O) + length(Y)] to null

end while

write line to studentrec

end while

close temp

close studentrec

declare variable X as integer

display "Go Back To Admin? [0 / 1] : "

input X

if X is equal to 0 then

call adminmenu()

else if X is equal to 1 then

display "Good Bye!"

end if

end function

### 2.3.12 Modify Programming Cafe Session Record Section

function modprgcafses()

display "Session\_code\tSession\_name\t\tWKday\t\tTime\tLocation\tTutor\_ID"

open file "Programming Cafe Session.txt" in read mode and assign it to prgsesrec

if prgsesrec is null then

display "Error opening file."

return

end if

display newline

declare variable line as string

while read a line from prgsesrec and assign it to line

display line

end while

close prgsesrec

declare variables O and Y as strings

display "Choose The Entry : "

input O

remove newline character from O

display "Replacement : "

input Y

remove newline character from Y

open file "out.txt" in write mode and assign it to temp

open file "Programming Cafe Session.txt" in read mode and assign it to prgsesrec

If temp or prgsesrec is null then

display "Error opening file."

return

end if

while read a line from prgsesrec and assign it to line

write line to temp

end while

close prgsesrec

close temp

open file "out.txt" in read mode and assign it to temp

open file "Programming Cafe Session.txt" in write mode and assign it to prgsesrec

if temp or prgsesrec is null then

display "Error opening file."

return

end if

while read a line from temp and assign it to line

declare variable pos as string

while pos is found by searching for O in line

set index to the position of pos in line

copy Y to pos

copy line starting from index + length(O) to the end to pos + length(Y)

set the character at line[length(line) - length(O) + length(Y)] to null

end while

write line to prgsesrec

end while

close temp

close prgsesrec

declare variable X as integer

display "Go Back To Admin? [0 / 1] : "

input X

if X is equal to 0 then

call adminmenu()

else if X is equal to 1 then

display "Good Bye!"

end if

end function

### 2.3.13 Enroll Student in a Session Section

function addstudses()

display "Participating Student in a Session Records"

open file "Enroll Student Session Records.txt" in append mode and assign it to studsesrec

if studsesrec is null then

display "Failed to open Enroll Student Session Records.txt"

return

end if

display "Enter Student & Session Details"

declare variables Name, TP\_Number, Prgsescode, Tutor\_ID, Location as strings

display "Enter Student Name: "

input Name

display "Enter TP Number: "

input TP\_Number

display "Enter Programming Session Code: "

input Prgsescode

display "Enter Tutor ID for this Session: "

input Tutor\_ID

display "Enter Location: "

input Location

write formatted data to studsesrec in the format "\n%s\t\t%s\t%s\t\t%s\t\t%s" with Name, TP\_Number, Prgsescode, Tutor\_ID, and Location as arguments

close studsesrec

declare variable X as integer

display "Do You Want To Continue? [0 / 1]: "

input X

if X is equal to 0 then

call adminmenu()

else if X is equal to 1 then

call addstudent()

else

display "Wrong Input"

call mainmenu()

end if

end function

## 2.4 Tutor Login Section

function tutorlogin()

declare tutorrec as file pointer

declare pass, Tutorid, str as strings

declare pos, pos1 as character pointers

declare count, count1, flag, flag1 as integers

display "Tutor Login Section"

open file "Tutor Records.txt" in read mode and assign it to tutorrec

if tutorrec is null then

display "Cannot Open File!!!"

else

display "Tutor ID : "

input Tutorid

display "Password : "

input pass

while (read a line into str from tutorrec)

set pos to the substring in str that matches Tutorid

if pos is not null then

if count is equal to 0 then

while (read a line into str from tutorrec)

set pos1 to the substring in str that matches pass

end while

if count1 is equal to 0 then

display "Login Successful!!"

close tutorrec

call tutormenu()

end if

set flag1 to 1

increment count1

end if

set flag to 1

increment count

end if

end while

end if

end function

## 2.5 Tutor Menu Section

function tutormenu()

declare tutoroption as integer

display "Welcome to Tutor Menu!"

display "Choose an option!"

display "1. Search & Display"

display " 11. Sessions Timing by Tutor ID"

display " 12. Participating Student Details by Tutor ID"

display "0. Exit"

declare tutorc1 as integer

display "Enter Your Choice: "

input tutorc1

switch (tutorc1)

case 11:

call distutorses()

break

case 12:

call disstudlist()

break

case 0:

call mainmenu()

break

default:

display "Invalid Choice"

call studentmenu()

end switch

end function

### 2.5.1 Search Session timing by Tutor ID Section

function distutorses()

open "Programming Cafe Session.txt" file for reading

if file does not exist or cannot be opened

display "Error opening file."

return

end if

declare sid as string

declare str as string

declare flag as integer

declare pos as character

display "Enter Tutor ID to Search : "

input sid

set flag to 0

while (read a line from prgsesrec and store it in str)

set pos to the result of searching sid in str

if pos is not NULL

display "Session\_code\tSession\_name\t\tWKday\t\tTime\tLocation\tTutor\_ID"

display str

set flag to flag + 1

exit while loop

end if

end while

if flag is 0

display "No Entry Found For Tutor\_ID : '<sid>' !!!"

end if

declare X as integer

display "Go Back To Admin? [0 / 1] : "

input X

if X is 0

call tutormenu()

else if X is 1

display "Good Bye!"

end if

end function

### 2.5.2 Search Participating Student List Section by Tutor ID

function disstudlist()

open "Enroll Student Session Records.txt" file for reading

if file does not exist or cannot be opened

display "Error opening file."

return

end if

declare sid as string

declare str as string

declare flag as integer

declare pos as character

display "Enter Tutor ID to Search : "

input sid

set flag to 0

while (read a line from studsesrec and store it in str)

set pos to the result of searching sid in str

if pos is not NULL

display "Student\_Name\tTP\_Number\tSession\_Code\tTutor\_ID\tLocation"

display str

set flag to flag + 1

exit while loop

end if

end while

if flag is 0

display "No Entry Found For Tutor\_ID : '<sid>' !!!"

end if

declare X as integer

display "Go Back To Admin? [0 / 1] : "

input X

if X is 0

call tutormenu()

else if X is 1

display "Good Bye!"

end if

end function

## 2.6 Student Login Section

function studentlogin()

open "Student Records.txt" file for reading

if file does not exist or cannot be opened

display "Cannot Open File!!!"

return

end if

declare pass as string

declare Tpnumber as string

declare str as string

declare pos as character pointer

declare pos1 as character pointer

declare count as integer

declare count1 as integer

declare flag as integer

declare flag1 as integer

display "Student Login Section"

display "TP Number : "

input Tpnumber

display "Password : "

input pass

set count to 0

set count1 to 0

set flag to 0

set flag1 to 0

while (read a line from studentrec and store it in str)

set pos to the result of searching Tpnumber in str

if pos is not NULL

if count is 0

while (read a line from studentrec and store it in str)

set pos1 to the result of searching pass in str

end while

if count1 is 0

declare match as integer

set match to the result of comparing str with str

if match is 0

display "Login Successful!!"

close studentrec

call studentmenu()

end if

end if

set flag1 to 1

increment count1 by 1

end if

set flag to 1

increment count by 1

end if

end while

end function

## 2.7 Student Menu Section

function studentmenu()

declare studentoption as integer

display "Welcome to Student Menu!"

display "Choose an option!"

display "1. Search & Display"

display " 11. Session Timing by TP Number"

display "0. Exit"

declare studentc1 as integer

display "Enter Your Choice: "

input studentc1

switch studentc1

case 11:

call disstudses()

break

case 0:

call mainmenu()

break

default:

display "Invalid Choice"

call studentmenu()

end switch

end function

### 2.7.1 Display Student Session Section

function disstudses()

declare studsesrec as file pointer

open "Enroll Student Session Records.txt" in read mode and assign it to studsesrec

if studsesrec is null

display "Error opening file."

return

end if

declare stid as character array of size 50

declare str as character array of size 120

declare flag as integer and initialize it to 0

declare pos as character pointer

display "Enter TP Number to Search : "

input stid

while fgets(str, 120, studsesrec) is not null

set pos to the result of strstr(str, stid)

if pos is not null

display "\n\nStudent\_Name\tTP\_Number\tSession\_Code\tTutor\_ID\tLocation\n\n"

display str

increment flag by 1

break the loop

end if

end while

if flag is 0

display "\nNo Entry Found For TP Number : 'stid' !!!"

end if

declare X as integer

display "\nGo Back To Admin? [0 / 1] : "

input X

if X is 0 then

call studentmenu()

else if X is 1 then

display "Good Bye!"

end if

end function

# 3.0 Additional Features

**Search & Display Specific Tutor Records**

A picture containing text, screenshot

Description automatically generated

**Search & Display Specific Student Records**

A picture containing text, screenshot, font, number

Description automatically generated

**Search & Display Specific Programming Café Session**

A picture containing text, screenshot, font, number

Description automatically generated

**Modify Tutor Records**

A screenshot of a computer program

Description automatically generated with medium confidence

**Modify Student Records**

A screenshot of a computer program

Description automatically generated with medium confidence

**Modify Programming Cafe Session Records**

A screenshot of a computer program

Description automatically generated with medium confidence

# 4.0 Sample Outputs

**Main Menu**

A picture containing text, screenshot, font

Description automatically generated

**Admin Login**

A picture containing text, screenshot, font

Description automatically generated

**Admin Menu**

A picture containing text, screenshot

Description automatically generated

**Add Tutor Information**

A picture containing text, screenshot, font

Description automatically generated

**Add Student Information**

A picture containing text, screenshot, font, black

Description automatically generated

**Add Programming Café Session Information**

A picture containing text, screenshot, font

Description automatically generated

**Display Tutor Records**



**Display Student Records**

A screenshot of a computer

Description automatically generated with medium confidence

**Display Programming Cafe Session Records**

A screenshot of a computer screen

Description automatically generated with medium confidence

**Search & Display Specific Tutor Records**

A black background with white text

Description automatically generated with low confidence

**Search & Display Specific Student Records**

A black background with white text

Description automatically generated with low confidence

**Search & Display Programming Cafe Session Records**

A black background with white text

Description automatically generated with low confidence

**Modify Tutor Records**

A screenshot of a computer program

Description automatically generated with medium confidence

**Modify Student Records**

A screenshot of a computer

Description automatically generated

**Modify Programming Cafe Session Records**

A screenshot of a computer

Description automatically generated with medium confidence

**Enroll Student in a Session**

A computer screen shot of white text

Description automatically generated with low confidence

**Tutor Login**

A screenshot of a computer screen

Description automatically generated with medium confidence

**Tutor Menu**

A black screen with white text

Description automatically generated with low confidence

**Search Session by Tutor ID**

A black background with white text

Description automatically generated with low confidence

**Search Student List by Tutor ID**

A screenshot of a computer program

Description automatically generated with low confidence

**Student Login**

A screen shot of a computer

Description automatically generated with medium confidence

**Student Menu**

A screen shot of a computer

Description automatically generated with low confidence

**Search Session Timing by TP Number**

A black screen with white text

Description automatically generated with low confidence

# 5.0 Conclusion

This project is made based on the C Programming Language giving a simple, reliable, and effective interface for a system service, spadework and development called “Programming Café”. Because C is a portable language and programs written in it may be readily compiled and executed on a wide range of systems. It facilitates controlling the program's sequence and providing logical results and write a C application that uses strings and stores several data kinds in the same memory.

# 6.0 References

* Thompson, B. (2023, May 6). GURU99. <https://www.guru99.com/c-programming-language.html>
* W3Schools. (n.d.). <https://www.w3schools.com/c/c_intro.php>
* *Pseudocode*. WIKIPEDIA. (n.d.). <https://en.wikipedia.org/wiki/Pseudocode>
* Programiz. (n.d.). <https://www.programiz.com/>