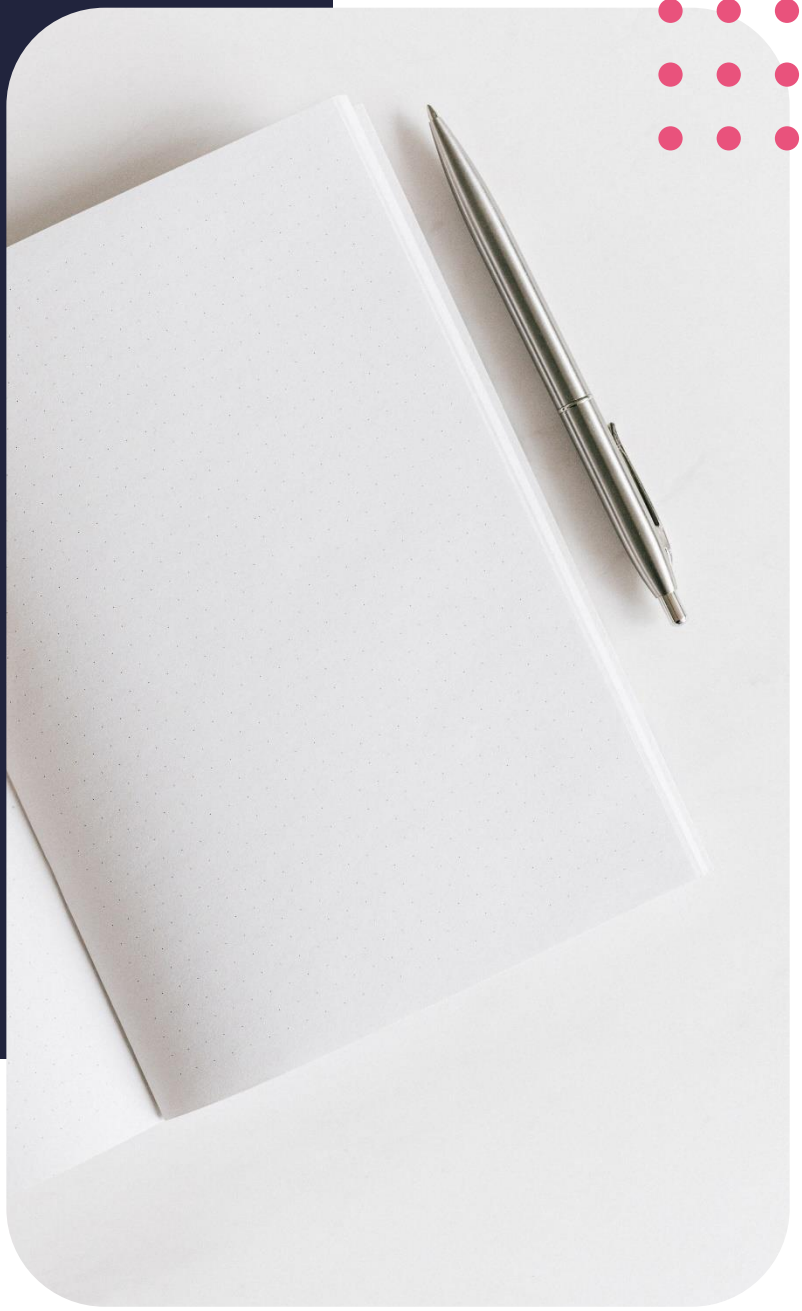
Diploma in Computer Science

Wrapping up

Contents



Lesson outcomes

Welcome to the last lesson in our course. we have come a long way and we have learned so much about computers! In this lesson we are primarily focused on putting all that we learned into practice. Most of all listen is going to be dedicated at finishing up our project and testing it out for the first time. We have written a bunch of functions that we're going to be putting together in order to make this happen.

Source code

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <string.h>

//Used macro

#define MAX_YR  9999

#define MIN_YR  1900

#define MAX_SIZE_USER_NAME 30

#define MAX_SIZE_PASSWORD  20

#define FILE_NAME  "studentRecordSystem.bin"

// Macro related to the students info

#define MAX_GUARDIAN_NAME 50

#define MAX_STUDENT_NAME 50

#define MAX_STUDENT_ADDRESS 300

#define FILE_HEADER_SIZE  sizeof(sFileHeader)

int handle=0;

//structure to store date

typedef struct

{

    int yyyy;

    int mm;
```

```
    int dd;
} Date;
typedef struct
{
    char username[MAX_SIZE_USER_NAME];
    char password[MAX_SIZE_PASSWORD];
} sFileHeader;
//Elements of structure
typedef struct// to call in program
{
    unsigned int student_id; // declare the integer data type
    char guardianName[MAX_GUARDIAN_NAME];// declare the character data type
    char studentName[MAX_STUDENT_NAME];// declare the character data type
    char studentAddr[MAX_STUDENT_ADDRESS];// declare the character data type
    Date studentJoiningDate;// declare the integer data type
} s_StudentInfo;
//Align the message
void printMessageCenter(const char* message)
{
    int len =0;
    int pos = 0;
    //calculate how many space need to print
    len = (78 - strlen(message))/2;
    printf("\t\t\t");
    for(pos =0 ; pos < len ; pos++)
    {
        //print space
        printf(" ");
    }
    //print message
```

```
    printf("%s",message);
}
//Head message
void headMessage(const char *message)
{
    system("cls");
welcome message


//Validate name
int isNameValid(const char *name)
{
    int validName = 1;
    checkif each character is valid
    for(index =0 to length of string)
    {
        if(!(isalpha(name[index])) && (name[index] != '\n') && (name[index] != ' '))
        {
            validName = 0;
            break;
        }
    }
    return validName;
}
// Function to check leap year.
//Function returns 1 if leap year
int  IsLeapYear(int year)
{
  If function is leap year
}
return 1
```

```
int isValidDate(Date *validDate)

{

    //check range of year,month and day

    if (validDate->yyyy > MAX_YR ||

            validDate->yyyy < MIN_YR)

        return 0;

    if (validDate->mm < 1 || validDate->mm > 12)

        return 0;

    if (validDate->dd < 1 || validDate->dd > 31)

        return 0;

    //Handle feb days in leap year

    if (validDate->mm == 2)

    {

        if (IsLeapYear(validDate->yyyy))

            return (validDate->dd <= 29);

        else

            return (validDate->dd <= 28);

    }

    //handle months which has only 30 days

    if (validDate->mm == 4 || validDate->mm == 6 ||

            validDate->mm == 9 || validDate->mm == 11)

        return (validDate->dd <= 30);

    return 1;

}

// Add student in list

void addStudentInDataBase()

{

    s_StudentInfo addStudentInfoInDataBase = {0};

    FILE *fp = NULL;

    FILE *fp1=NULL;
```

```
    int status = 0;
    fp = fopen(FILE_NAME,"ab+");
    if(fp == NULL)
    {
        printf("File isn't open\n");
        exit(1);
    }
    headMessage("ADD NEW Students");
    printf("\n\n\t\t\tENTER YOUR DETAILS BELOW:");
printf("\n\t\t\t===============================================================================\n");
    printf("\n\t\t\tStudent ID  = ");
    fflush(stdin);
    scanf("%u",&addStudentInfoInDataBase.student_id);
    handle=addStudentInfoInDataBase.student_id;
    ////////
  dupeCheck();
    ////////
    do
    {
        printf("\n\t\t\tGuardian Name  = ");
        fflush(stdin);
        fgets(addStudentInfoInDataBase.guardianName,MAX_GUARDIAN_NAME,stdin);
        status = isNameValid(addStudentInfoInDataBase.guardianName);
        if (!status)
        {
            printf("\n\t\t\tName contains an invalid character. Please enter again.");
        }
    }
    while(!status);
    do
```

```
    {
        printf("\n\t\t\tStudent Name  = ");
        fflush(stdin);
        fgets(addStudentInfoInDataBase.studentName,MAX_STUDENT_NAME,stdin);
        status = isNameValid(addStudentInfoInDataBase.studentName);
        if (!status)
        {
            printf("\n\t\t\tName contains an invalid character. Please enter again.");
        }
    }
    while(!status);
    do
    {
        printf("\n\t\t\tStudent Address  = ");
        fflush(stdin);
        fgets(addStudentInfoInDataBase.studentAddr,MAX_GUARDIAN_NAME,stdin);
        status = isNameValid(addStudentInfoInDataBase.studentAddr);
        if (!status)
        {
            printf("\n\t\t\tName contains an invalid character. Please enter again.");
        }
    }
    while(!status);
    do
    {
        //get date year,month and day from user
        printf("\n\t\t\tEnter date in format (day/month/year): ");

scanf("%d/%d/%d",&addStudentInfoInDataBase.studentJoiningDate.dd,&addStudentInfoInDataBase.studentJoiningDate
.mm,&addStudentInfoInDataBase.studentJoiningDate.yyyy);

        //check date validity
```

```
        status = isValidDate(&addStudentInfoInDataBase.studentJoiningDate);
        if (!status)
        {
            printf("\n\t\t\tPlease enter a valid date.\n");
        }
    }
    while(!status);
    fwrite(&addStudentInfoInDataBase,sizeof(addStudentInfoInDataBase), 1, fp);
    fclose(fp);
}
// search student
void searchStudent()
{
    int found = 0;
    int studentId =0;
    s_StudentInfo addStudentInfoInDataBase = {0};
    FILE *fp = NULL;
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
        printf("\n\t\t\tFile isn't open\n");
        exit(1);
    }
    headMessage("SEARCH STUDENTS");
    //put the control on student detail
    if (fseek(fp,FILE_HEADER_SIZE,SEEK_SET) != 0)
    {
        fclose(fp);
        printf("\n\t\t\tWe're having trouble reading the file\n");
        exit(1);
```

```
    }

    printf("\n\n\t\t\tEnter Student ID NO to search:");

    fflush(stdin);

    scanf("%u",&studentId);

    while (fread (&addStudentInfoInDataBase, sizeof(addStudentInfoInDataBase), 1, fp))

    {

        if(addStudentInfoInDataBase.student_id == studentId)

        {

            found = 1;

            break;

        }

    }

    if(found)

    {

        printf("\n\t\t\tStudent id = %d\n",addStudentInfoInDataBase.student_id);

        printf("\n\t\t\tStudent name = %s",addStudentInfoInDataBase.studentName);

        printf("\t\t\tGuardian Name = %s",addStudentInfoInDataBase.guardianName);

        printf("\n\t\t\tStudent Address = %s",addStudentInfoInDataBase.studentAddr);

        printf("\t\t\tStudent                 Admission                Date(day/month/year)                =
(%d/%d/%d)",addStudentInfoInDataBase.studentJoiningDate.dd,

               addStudentInfoInDataBase.studentJoiningDate.mm,
addStudentInfoInDataBase.studentJoiningDate.yyyy);

    }

    else

    {

        printf("\n\t\t\tNo Record");

    }

    fclose(fp);

    printf("\n\n\n\t\t\tPress any key to go to main menu.");

    fflush(stdin);

    getchar();
```

```
}

// view students function

void viewStudent()

{

    int found = 0;

    s_StudentInfo addStudentInfoInDataBase = {0};

    FILE *fp = NULL;

    unsigned int countStudent = 1;

    headMessage("VIEW STUDENT DETAILS");

    fp = fopen(FILE_NAME,"rb");

    if(fp == NULL)

    {

        printf("File isn't open\n");

        exit(1);

    }

    if (fseek(fp,FILE_HEADER_SIZE,SEEK_SET) != 0)

    {

        fclose(fp);

        printf("We're having trouble reading the file\n");

        exit(1);

    }

    while (fread (&addStudentInfoInDataBase, sizeof(addStudentInfoInDataBase), 1, fp))

    {

        printf("\n\t\t\tStudent Count = %d\n\n",countStudent);

        printf("\t\t\tStudent id = %u\n",addStudentInfoInDataBase.student_id);

        printf("\t\t\tStudent Name = %s",addStudentInfoInDataBase.studentName);

        printf("\t\t\tGuardian Name = %s",addStudentInfoInDataBase.guardianName);

        printf("\t\t\tStudent Address = %s",addStudentInfoInDataBase.studentAddr);

        printf("\t\t\tStudent                Admission                Date(day/month/year)                =
(%d/%d/%d)\n\n",addStudentInfoInDataBase.studentJoiningDate.dd,
```

```
                addStudentInfoInDataBase.studentJoiningDate.mm,
addStudentInfoInDataBase.studentJoiningDate.yyyy);

        found = 1;

        ++countStudent;

    }

    fclose(fp);

    if(!found)

    {

        printf("\n\t\t\tNo Record");

    }

    printf("\n\n\t\t\tPress any key to go to main menu.");

    fflush(stdin);

    getchar();

}

// Delete student entry

void deleteStudent()

{

    int found = 0;

    int studentDelete = 0;

    sFileHeader fileHeaderInfo = {0};

    s_StudentInfo addStudentInfoInDataBase = {0};

    FILE *fp = NULL;

    FILE *tmpFp = NULL;

    headMessage("Delete Student Details");

    fp = fopen(FILE_NAME,"rb");

    if(fp == NULL)

    {

        printf("File isn't open\n");

        exit(1);

    }
```

```
    tmpFp = fopen("tmp.bin","wb");
    if(tmpFp == NULL)
    {
        fclose(fp);
        printf("File isn't open\n");
        exit(1);
    }
    fread (&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
    fwrite(&fileHeaderInfo,FILE_HEADER_SIZE, 1, tmpFp);
    printf("\n\t\t\tEnter Student ID NO. for delete:");
    scanf("%d",&studentDelete);
    while (fread (&addStudentInfoInDataBase, sizeof(addStudentInfoInDataBase), 1, fp))
    {
        if(addStudentInfoInDataBase.student_id != studentDelete)
        {
            fwrite(&addStudentInfoInDataBase,sizeof(addStudentInfoInDataBase), 1, tmpFp);
        }
        else
        {
            found = 1;
        }
    }
    (found)? printf("\n\t\t\tRecord deleted successfully."):printf("\n\t\t\tRecord not found");
    getch();
    fclose(fp);
    fclose(tmpFp);
    remove(FILE_NAME);
    rename("tmp.bin",FILE_NAME);
}
//function to update credential
```

```
void updateCredential(void)

{

    sFileHeader fileHeaderInfo = {0};

    FILE *fp = NULL;

    unsigned char userName[MAX_SIZE_USER_NAME] = {0};

    unsigned char password[MAX_SIZE_PASSWORD] = {0};

    headMessage("Update Credentials");

    fp = fopen(FILE_NAME,"rb+");

    if(fp == NULL)

    {

        printf("File isn't open\n");

        exit(1);

    }

    fread (&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);

    if (fseek(fp,0,SEEK_SET) != 0)

    {

        fclose(fp);

        printf("\n\t\t\tWe're having trouble updating the password\n");

        exit(1);

    }

    printf("\n\n\t\t\tNew Username:");

    fflush(stdin);

    fgets(userName,MAX_SIZE_USER_NAME,stdin);

    printf("\n\n\t\t\tNew Password:");

    fflush(stdin);

    fgets(password,MAX_SIZE_PASSWORD,stdin);

    strncpy(fileHeaderInfo.username,userName,sizeof(userName));

    strncpy(fileHeaderInfo.password,password,sizeof(password));

    fwrite(&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);

    fclose(fp);
```

```
    printf("\n\t\t\tYour Password has been changed successfully");
    printf("\n\t\t\ttLogin Again:");
    fflush(stdin);
    getchar();
    exit(1);
}
//Display menu
void menu()
{
    int choice = 0;
    do
    {
        headMessage("MAIN MENU");
        printf("\n\n\n\t\t\t1.Add Student");
        printf("\n\t\t\t2.Search Student");
        printf("\n\t\t\t3.View Student");
        printf("\n\t\t\t4.Delete Student");
        printf("\n\t\t\t5.Update Password");
        printf("\n\t\t\t0.Exit");
        printf("\n\n\n\t\t\tEnter choice => ");
        scanf("%d",&choice);
        switch(choice)
        {
        case 1:
            addStudentInDataBase();
            break;
        case 2:
            searchStudent();
            break;
        case 3:
```

```
            viewStudent();
            break;
        case 4:
            deleteStudent();
            break;
        case 5:
            updateCredential();
            break;
        case 0:
            printf("\n\n\n\t\t\t\tThank you!!!\n\n\n\n\n");
            exit(1);
            break;
        default:
            printf("\n\n\n\t\t\tINVALID INPUT!!! Please try again...");
        }                                           //Switch Ended
    }
    while(choice!=0);                                       //Loop Ended
}
//login password
void login()
{
    unsigned char userName[MAX_SIZE_USER_NAME] = {0};
    unsigned char password[MAX_SIZE_PASSWORD] = {0};
    int L=0;
    sFileHeader fileHeaderInfo = {0};
    FILE *fp = NULL;
    headMessage("Login");
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
```

```
        printf("File isn't open\n");
        exit(1);
    }
    fread (&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
    fclose(fp);
    do
    {
        printf("\n\n\n\t\t\t\tUsername:");
        fgets(userName,MAX_SIZE_USER_NAME,stdin);
        printf("\n\t\t\t\tPassword:");
        fgets(password,MAX_SIZE_PASSWORD,stdin);
        if((!strcmp(userName,fileHeaderInfo.username)) && (!strcmp(password,fileHeaderInfo.password)))
        {
            menu();
        }
        else
        {
            printf("\t\t\t\tLogin Failed, please check your Username & Password\n\n");
            L++;
        }
    }
    while(L<=3);
    if(L>3)
    {
        headMessage("Login Failed");
        printf("\t\t\t\tSorry, too many login attempts. Goodbye!");
        getch();
        system("cls");
    }
}
```

```
//Check file exist or not

int isFileExists(const char *path)

{

    // Try to open file

    FILE *fp = fopen(path, "rb");

    int status = 0;

    // If file does not exists

    if (fp != NULL)

    {

        status = 1;

        // File exists hence close file

        fclose(fp);

    }

    return status;

}

void init()

{

    FILE *fp = NULL;

    int status = 0;

    const char defaultUsername[] ="shaw\n";

    const char defaultPassword[] ="academy\n";

    sFileHeader fileHeaderInfo = {0};

    status = isFileExists(FILE_NAME);

    if(!status)

    {

        //create the binary file

        fp = fopen(FILE_NAME,"wb");

        if(fp != NULL)

        {

            //Copy default password
```

```
            strncpy(fileHeaderInfo.password,defaultPassword,sizeof(defaultPassword));
            strncpy(fileHeaderInfo.username,defaultUsername,sizeof(defaultUsername));
            fwrite(&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
            fclose(fp);
        }
    }
}
// duplicate check
void dupeCheck()
{
 int found = 0;
    int studentId =0;
    s_StudentInfo addStudentInfoInDataBase = {0};
    FILE *fp = NULL;
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
        printf("\n\t\t\tFile isn't open\n");
        exit(1);
    }
       if (fseek(fp,FILE_HEADER_SIZE,SEEK_SET) != 0)
    {
        fclose(fp);
        printf("\n\t\t\tWe're having trouble reading the file\n");
        exit(1);
    }
        fflush(stdin);
        studentId=handle;
       while (fread (&addStudentInfoInDataBase, sizeof(addStudentInfoInDataBase), 1, fp))
    {
```

```
        if(addStudentInfoInDataBase.student_id == studentId)

        {

            found = 1;

            break;

        }

    }

    if(found)

    {

        printf("\n\t\t\tThis record exists");

        printf("\n\t\t\tStudent id = %d\n",addStudentInfoInDataBase.student_id);

        printf("\n\t\t\tStudent name = %s",addStudentInfoInDataBase.studentName);

        printf("\t\t\tGuardian Name = %s",addStudentInfoInDataBase.guardianName);

        printf("\n\t\t\tStudent Address = %s",addStudentInfoInDataBase.studentAddr);

        printf("\t\t\tStudent                 Admission                Date(day/month/year)                =
(%d/%d/%d)",addStudentInfoInDataBase.studentJoiningDate.dd,

               addStudentInfoInDataBase.studentJoiningDate.mm,
addStudentInfoInDataBase.studentJoiningDate.yyyy);

        printf("\n\t\t\tPlease use a unique record number");

        getch();

        menu();

    }

    else

    {

        return;

    }

    fclose(fp);

    printf("\n\n\n\t\t\tPress any key to go to main menu.");

    fflush(stdin);

    getchar();

}

int main()
```

```
{
    init();
    welcomeMessage();
    login();
    return 0;
}
```