



INFORMATICS
INSTITUTE OF
TECHNOLOGY

Foundation Certificate in Higher Education.

Module : DOC 334 Introduction To Programming P2

Module Leader : Mr. Sudharshana Welihindha

Assessment Type : Individual Coursework

Student ID : 20211490

Table Of Content

Problem Statement.....	1
Solution.....	1
Screenshots Of Test Cases.....	2
Test Cases Summary.....	14
Code.....	26

List Of Tables

Table 1 Test Case 1.....	2
Table 2 Test Case 2.....	3
Table 3 Test Case 3.....	4
Table 4 Test Case 4.....	5
Table 5 Test Case 5.....	6
Table 6 Test Case 6.....	7
Table 7 Test Case 7.....	8
Table 8 Test Case 8.....	9
Table 9 Test Case 9.....	10
Table 10 Test Case 10.....	11
Table 11 Test Case 11	11
Table 12 Test Case 12.....	12
Table 13 Test Case 13.....	13

List Of Figures

Figure 1 Test Case 1.....	14
Figure 2 Test Case 2.....	15
Figure 3 Test Case 3.....	16
Figure 4 Test Case 4.....	16
Figure 5 Test Case 5.....	17
Figure 6 Test Case 6.....	18
Figure 7 Test Case 7.....	19
Figure 8 Test Case 8.....	20
Figure 9 Test Case 9.....	21
Figure 10 Test Case 10.....	22
Figure 11 Test Case 11.....	23
Figure 12 Test Case 12.....	24
Figure 13 Test Case 13.....	25

Problem statement

Absence of a proper method of storing the details about students and their attendance for small classrooms have affected the progress of those classes badly. Even though it could still be done manually, it will not sound that efficient. There might be a chance to lose those records, also difficult while retrieving data and when it's needed to update some records, it's getting way too difficult. It is needed to develop a console-based application to avoid these disturbances.

Solution

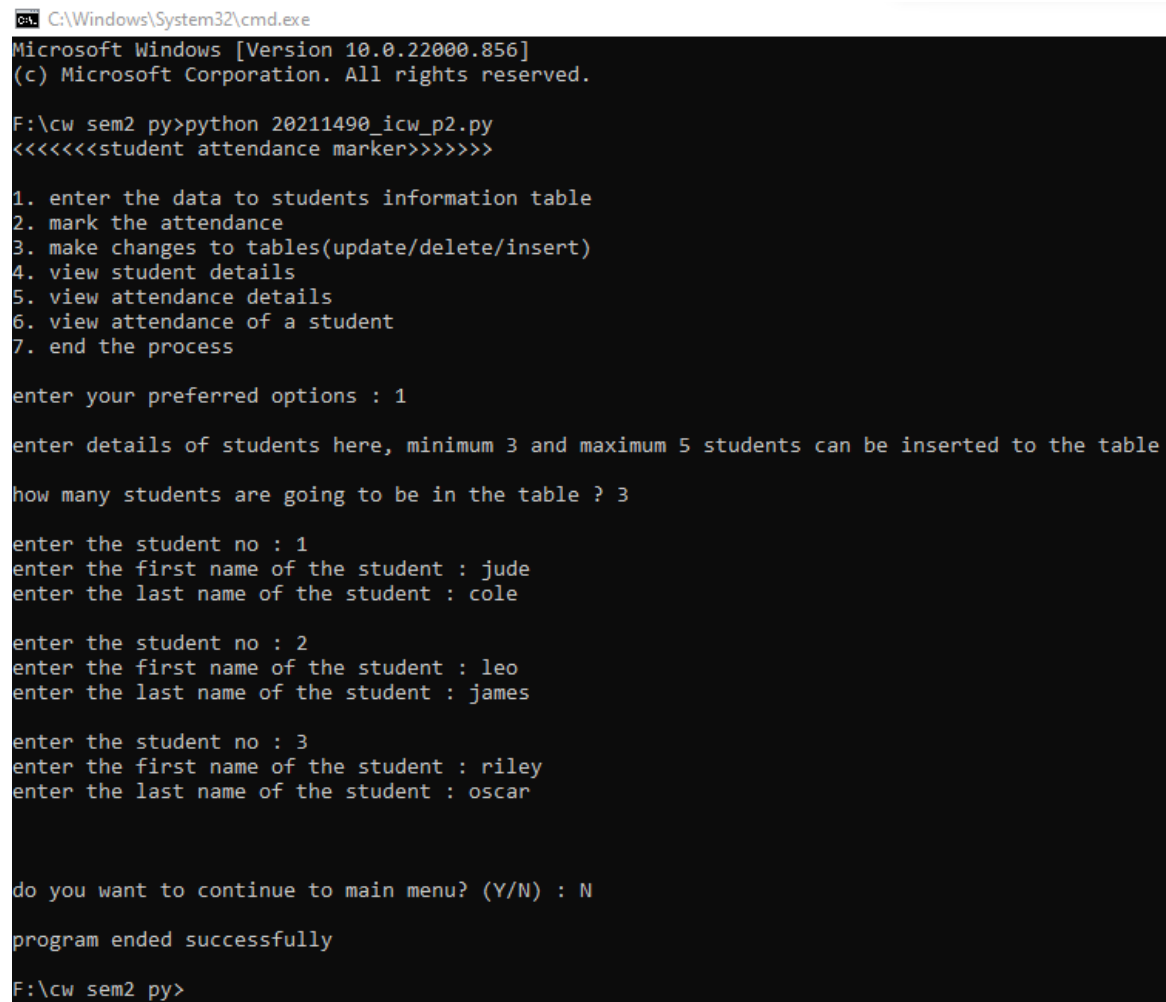
The developed program will help teachers to keep track of the details of the students and their attendance. This program is basically developed using python and MySQL.

2 tables are created in MySQL under a server called "cls". 1st table is created to store details of students like student number, first name and last name of students. 2nd table is keeping track of the attendance of each student. And since it's made for small classrooms, the limit is brought to maximum 5 records and minimum 3 records.

The user is also allowed to make changes like, inserting new records, updating existing records and deleting records. the user is always able to see view the records of both tables as well as the attendance of a selected student. Error handling is playing a great role during the process, it will let the user know what mistake they have done, and the user can work accordingly.

Screenshots of Test Cases

Test Case 1



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

F:\cw sem2 py>python 20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 1

enter details of students here, minimum 3 and maximum 5 students can be inserted to the table

how many students are going to be in the table ? 3

enter the student no : 1
enter the first name of the student : jude
enter the last name of the student : cole

enter the student no : 2
enter the first name of the student : leo
enter the last name of the student : james

enter the student no : 3
enter the first name of the student : riley
enter the last name of the student : oscar

do you want to continue to main menu? (Y/N) : N

program ended successfully

F:\cw sem2 py>
```

Figure 1

Test Case 2

```
C:\Windows\System32\cmd.exe
F:\cw sem2 py>python 20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 2

enter the date in dd-mm-yy format : 18-08-2022

enter the student number displayed above and then mark the attendance(P if present,A if absent)
(1,)

student No : 1
enter P if present,A if absent : P

enter the student number displayed above and then mark the attendance(P if present,A if absent)
(2,)

student No : 2
enter P if present,A if absent : P

enter the student number displayed above and then mark the attendance(P if present,A if absent)
(3,)

student No : 3
enter P if present,A if absent : A

do you want to continue to main menu? (Y/N) : N

program ended successfully

F:\cw sem2 py>
```

Figure 2

Test Case 3

```
F:\cw sem2 py>python 20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 3

what change you want to do?
1. update
2. insert
3. delete
enter your preferred option : 1
what you want to update?

1. update student number
2. update first name
3. update last name
what operation you need to do : 1
enter the first name of the student that you need to change the student number of : leo
enter the new student number : 4

do you want to continue to main menu? (Y/N) : N

program ended successfully

F:\cw sem2 py>
```

Figure 3

Test Case 4

```
F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 3

what change you want to do?
1. update
2. insert
3. delete
enter your preferred option : 1
what you want to update?

1. update student number
2. update first name
3. update last name
what operation you need to do : 2
enter the student number of the student that you need to change the student firstname of : 3
enter the new first name of the student : rhett

do you want to continue to main menu? (Y/N) : N

program ended successfully

F:\cw sem2 py>
```

Figure 4

Test Case 5

```
F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 3

what change you want to do?
1. update
2. insert
3. delete
enter your preferred option : 1
what you want to update?

1. update student number
2. update first name
3. update last name
what operation you need to do : 3
enter the studnet number of the student that you need to change the student lastname of : 3
enter the new last name of the student : andrew

do you want to continue to main menu? (Y/N) : n

program ended successfully

F:\cw sem2 py>
```

Figure 5

Test Case 6

```
F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 3

what change you want to do?
1. update
2. insert
3. delete
enter your preferred option : 2
enter the student no : 5
enter the first name of the student : walker
enter the last name of the student : william

1 record added

do you want to continue to main menu? (Y/N) : N

program ended successfully

F:\cw sem2 py>
```

Figure 6

Test Case 7

```
F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 3

what change you want to do?
1. update
2. insert
3. delete
enter your preferred option : 3
what record you want to delete?
student No   : date : attendance

(1, 'jude', 'cole')
(3, 'rhett', 'andrew')
(4, 'leo', 'james')
(5, 'walker', 'william')

type the student number of the record that you need to delete : 1

1 record deleted successfully

do you want to continue to main menu? (Y/N) : N

program ended successfully

F:\cw sem2 py>
```

Figure 7

Test Case 8

```
F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 4

student No : first name : last name
(3, 'rhett', 'andrew')
(4, 'leo', 'james')
(5, 'walker', 'william')
do you want to continue to main menu? (Y/N) : N

program ended successfully

F:\cw sem2 py>
```

Figure 8

Test Case 9

```
F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 5

student No   : date : attendance

(1, '18-08-2022', 'P')
(2, '18-08-2022', 'P')
(3, '18-08-2022', 'A')

do you want to continue to main menu? (Y/N) : N

program ended successfully

F:\cw sem2 py>
```

Figure 9

Test Case 10

```
F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 6

enter the student number : 3

student No : date : attendance

[(3, '18-08-2022', 'A')]

do you want to continue to main menu? (Y/N) :
```

Figure 10

Test Case 11

```
F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 7

left from the student attendance marker successfully!

F:\cw sem2 py>
```

Figure 11

Test case 12

```
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 8

please enter a valid input
do you want to continue to main menu? (Y/N) : Y
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options :
```

Figure 12

Test Case 13

```
F:\cw sem2 py>20211490_icw_p2.py
<<<<<<student attendance marker>>>>>>

1. enter the data to students information table
2. mark the attendance
3. make changes to tables(update/delete/insert)
4. view student details
5. view attendance details
6. view attendance of a student
7. end the process

enter your preferred options : 1

enter details of students here, minimum 3 and maximum 5 students can be inserted to the table

how many students are going to be in the table ? 7

please make sure to enter at least 3 records and maximum 5 records

do you want to continue to main menu? (Y/N) : N

program ended successfully
```

Figure 13

Test Cases Summary

Test Case 1

Table 1

Inputs: Preferred option: 1 How many records: 3 Student number(s): 1,2,3 First names: jude, leo, riley Last names: cole, james, oscar
Expected output: do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output: do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test Case 2

Table 2

Inputs: Preferred option: 2 Date: 18-08-2022 Student number :1,2,3 Present/absent : P,P,A
Expected output: do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output: do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test Case 3

Table 3

Inputs: Preferred option: 3 Preferred mini option(update/delete/insert): 1 Preferred mini option (update student number/first name/last name) : 1 first name of the student that you need to change the student number of : leo new student number : 4
Expected output: do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output: do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test Case 4

Table 4

Inputs: Preferred option: 3 Preferred mini option(update/delete/insert): 1 Preferred mini option (update student number/first name/last name) : 2 Student number of the student that you need to change the first name of : 3 new first name of the student : rhett
Expected output: do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output: do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test Case 5

Table 5

Inputs: Preferred option: 3 Preferred mini option(update/delete/insert): 1 Preferred mini option (update student number/first name/last name) : 3 Student number of the student that you need to change the last name of : 3 new last name of the student : andrew
Expected output: do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output: do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test case 6

Table 6

Inputs: Preferred option: 3 Preferred mini option(update/delete/insert): 2 Student number : 5 First name: walker Last name: william
Expected output: 1 record added do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output: 1 record added do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test case 7

Table 7

Inputs: Preferred option: 3 Student number of the record that is needed to be deleted: 1
Expected output: what record you want to delete? student No : date : attendance (1, 'jude', 'cole') (3, 'rhett', 'andrew') (4, 'leo', 'james') (5, 'walker', 'william') type the student number of the record that you need to delete : 1 1 record deleted successfully do you want to continue to main menu? (Y/N) : n program ended successfully
Actual output : what record you want to delete? student No : date : attendance (1, 'jude', 'cole') (3, 'rhett', 'andrew') (4, 'leo', 'james') (5, 'walker', 'william') type the student number of the record that you need to delete : 1 1 record deleted successfully do you want to continue to main menu? (Y/N) : n program ended successfully
Remarks: Test Case Passed

Test case 8

Table 8

Inputs: Preferred option: 4
Expected output: student No : first name : last name (3, 'rhett', 'andrew') (4, 'leo', 'james') (5, 'walker', 'william') do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output: student No : first name : last name (3, 'rhett', 'andrew') (4, 'leo', 'james') (5, 'walker', 'william') do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test case 9

Table 9

Inputs: Preferred option: 5
Expected output : student No : date : attendance (1, '18-08-2022', 'P') (2, '18-08-2022', 'P') (3, '18-08-2022', 'A') do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output : student No : date : attendance (1, '18-08-2022', 'P') (2, '18-08-2022', 'P') (3, '18-08-2022', 'A') do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test Case 10

Table 10

Inputs: Preferred option: 6 Student number : 3
Expected output : student No : date : attendance [(3, '18-08-2022', 'A')] do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output : student No : date : attendance [(3, '18-08-2022', 'A')] do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test Case 11

Table 11

Inputs: Preferred option: 7
Expected output: left from the student attendance marker successfully!
Actual output: left from the student attendance marker successfully!
Remarks: Test Case Passed

Test Case 12

Table 12

Inputs: Preferred option: 1 How many records: 7
Expected output: please make sure to enter at least 3 records and maximum 5 records do you want to continue to main menu? (Y/N) : N program ended successfully
Actual output: please make sure to enter at least 3 records and maximum 5 records do you want to continue to main menu? (Y/N) : N program ended successfully
Remarks: Test Case Passed

Test Case 13

Table 13

Inputs: Preferred option: 8
Expected output: please enter a valid input do you want to continue to main menu? (Y/N) : Y <<<<<<student attendance marker>>>>>> 1. enter the data to students information table 2. mark the attendance 3. make changes to tables(update/delete/insert) 4. view student details 5. view attendance details 6. view attendance of a student 7. end the process
Actual output: please enter a valid input do you want to continue to main menu? (Y/N) : Y <<<<<<student attendance marker>>>>>> 1. enter the data to students information table 2. mark the attendance 3. make changes to tables(update/delete/insert) 4. view student details 5. view attendance details 6. view attendance of a student 7. end the process
Remarks: Test Case Passed

Code

```
import mysql.connector #connecting python with MySQL

#exception handling
try:
    conDict ={'host':'localhost',
              'user':'root',
              'database':'miniclass',
              'password':"",
              'connect_timeout':1000 }
    db = mysql.connector.connect(**conDict) #creating a database connection
    cursor = db.cursor(buffered = True) #creating a cursor

except Exception:
    print("""OOPS!!
    seems like something's wrong with your MySQL database connection,
    please try again before starting the following process""")
    print()
    print()
    print()
    print()

#functions of the program
```

```

def main():
    print("<<<<<<student attendance marker>>>>>>") #this function will return the main
    menu of the program

    print()
    print("1. enter the data to students information table")
    print("2. mark the attendance")
    print("3. make changes to tables(update/delete/insert)")
    print("4. view student details")
    print("5. view attendance details")
    print("6. view attendance of a student")
    print("7. end the process")
    print()
    global choice
    choice = int(input("enter your preferred options : "))
    print()

def studentIn(): #this function will get records from user to store in the studentstbl
    print("enter details of students here, minimum 3 and maximum 5 students can be inserted to
    the table")

    print()
    record_no = int(input("how many students are going to be in the table ? "))
    print()
    if record_no >2 and record_no <=5:
        for no in range (record_no):
            global studentNo
            studentNo = int(input("enter the student no : "))
            firstName = input("enter the first name of the student : ")

```

```

        lastName = input("enter the last name of the student : ")
        print()
        mySQLText = ("INSERT INTO studentstbl (studentNo,firstname,lastname) VALUES
(%s,%s,%s)")
        student_details = (studentNo,firstName,lastName)
        cursor.execute(mySQLText,student_details)
    else:
        print()
        print("please make sure to enter at least 3 records and maximum 5 records")

```

```

def attendanceInfo(): #this function will mark the attendace with the records from user
    date_in = input("enter the date in dd-mm-yy format : ")
    print()
    cursor.execute("SELECT studentNo FROM studentstbl")
    data = cursor.fetchall()

```

```

    for item in data:
        print("enter the student number displayed above and then mark the attendance(P if
present,A if absent)")
        print()
        print(item)
        print()
        studentNo = int(input("student No : "))
        attendance = input("enter P if present,A if absent : ").upper()
        print()

```



```
mycmd = "INSERT INTO attendancetbl (studentNo,attendance,date) VALUES (%s,%s,%s) "
```

```
det = (studentNo,attendance,date_in)
```

```
cursor.execute(mycmd,det)
```

```
def update(): # a function which contains a small menu which comes under update option
```

```
    print("what you want to update?")
```

```
    print()
```

```
    print("1. update student number")
```

```
    print("2. update first name ")
```

```
    print("3. update last name")
```

```
def stunoupd(): #this function will update the student number
```

```
    info = input("enter the first name of the student that you need to change the student number  
of : ")
```

```
    newstuNo = input("enter the new student number : ")
```

```
    uptxt = "UPDATE studentstbl SET studentNo = '{}' WHERE firstName =  
'{}'".format(newstuNo,info)
```

```
    cursor.execute(uptxt)
```

```
def stuFirstNameupd(): #this function will update the first name of students
```

```
info = input("enter the student number of the student that you need to change the student  
firstname of : ")
```

```
newstuName = input("enter the new first name of the student : ")
```

```
uptxt = "UPDATE studentstbl SET firstName = '{}' WHERE studentNo =  
'{}'".format(newstuName,info)
```

```
cursor.execute(uptxt)
```

```
def stuLastNameupd(): #this function will update the last name of students
```

```
info = input("enter the student number of the student that you need to change the student  
lastname of : ")
```

```
newstuName = input("enter the new last name of the student : ")
```

```
uptxt = "UPDATE studentstbl SET lastName = '{}' WHERE studentNo =  
'{}'".format(newstuName,info)
```

```
cursor.execute(uptxt)
```

```
def stuInput(): #this function will let the user to input new records to studentstbl
```

```
studentNo = input("enter the student no : ")
```

```
firstName = input("enter the first name of the student : ")
```

```
lastName = input("enter the last name of the student : ")
```

```
print()
```

```
mySQLText = ("INSERT INTO studentstbl (studentNo,firstname,lastname) VALUES  
(%s,%s,%s)")
```

```
student_details = (studentNo,firstName,lastName)
```

```
cursor.execute(mySQLText,student_details)
```

```

def delete(): #this function will let the users to delete records from studentstbl

    print("what record you want to delete?")
    cursor.execute ("SELECT * FROM studentstbl")
    data = cursor.fetchall()

    print("student No : date : attendance")
    print()
    for item in data: #printing the records one by one with for loop
        print(item)
    print()
    delRec = input("type the student number of the record that you need to delete : ")
    cursor.execute("DELETE FROM studentstbl WHERE studentNo = "+ delRec + "")
    print()
    print(cursor.rowcount, "record deleted successfully")

```

```

try: #exception hanfling
    main() #calling main function
except Exception as e:
    print("OOPS! somethings's wrong")
    print(e)
while True: #while loop getting started

    if choice == 1:
        try: #exception handling
            studentIn() #calling studentIn function

```

```

db.commit() #saving changes

print()

print()

need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper() #asking
whether the user needs to continue or not

if need_to_cont == "Y":

    main()

elif need_to_cont == "N":

    db.close() #disconnecting from MySQL server

    print()

    print("program ended successfully")

    break

else:

    db.close() #disconnecting from MySQL server

    print("invalid input")

    break

except Exception as e:

    print("OOPS! something went wrong,please try again") #error message

    print(e)

    print()

    main() #calling main function


elif choice == 2:

    try: #exception handling

```

```

attendanceInfo() #calling attendanceInfo function

db.commit() #saving changes

print()

print()

need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper()
#asking whether the user needs to continue or not

if need_to_cont == "Y":

    main()

elif need_to_cont == "N":

    db.close()

    print()

    print("program ended successfully")

    break

else:

    db.close()

    print("invalid input")

    break

except Exception as e:

    print("OOPS! something went wrong,please try again") #error message

    print(e)

    print()

    main() #calling main function


elif choice == 3:

    print("what change you want to do?") #a small menu for 3rd choice

    print("1. update")

```

```
print("2. insert")
```

```
print("3. delete")
```

```
try: #exception handling
```

```
mychoice = int(input("enter your preferred option : "))
```

```
if mychoice == 1:
```

```
    update() #calling update function
```

```
    upt = int(input("what operation you need to do : ")) #getting user input to proceed
```

```
    if upt == 1:
```

```
        stunoupd() #calling stunoupd function to update student's student number
```

```
        db.commit() #saving changes
```

```
        print()
```

```
        print()
```

```
        need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper()
```

```
#asking whether the user needs to continue or not
```

```
        if need_to_cont == "Y":
```

```
            main() #calling the main menu if user needs to continue
```

```
        elif need_to_cont == "N":
```

```
            db.close() #disconnecting from MySQL server
```

```
            print()
```

```
            print("program ended successfully")
```

```
            break
```

```
        else:
```

```
            db.close()
```

```
            print("invalid input")
```

```
            break
```

```

elif upt == 2:

    stuFirstNameupd() #calling stuNameupd function to update student's first name

    db.commit()

    print()

    print()

    need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper()
#asking whether the user needs to continue or not

    if need_to_cont == "Y":

        main() #calling main function

    elif need_to_cont == "N":

        db.close()

        print()

        print("program ended successfully")

        break

    else:

        db.close()

        print()

        print("invalid input")

        break

elif upt == 3:

    stuLastNameupd() #calling stuNameupd function to update student's last name

    db.commit() #saving changes

    print()

    print()

    need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper()
#asking whether the user needs to continue or not

    if need_to_cont == "Y":

        main()

```

```

elif need_to_cont == "N":
    db.close()
    print()
    print("program ended successfully")
    break
else:
    db.close()
    print()
    print("invalid input")
    break
elif mychoice == 2:

    stuInput() #calling stuNoIn function to insert new data records
    db.commit()
    print(cursor.rowcount, "record added")
    print()
    print()

    need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper()
#asking whether the user needs to continue or not
    if need_to_cont == "Y":
        main()
    elif need_to_cont == "N":
        db.close()
        print()
        print("program ended successfully")
        break
    else:

```



```

        db.close()

        print()

        print("invalid input")

        break

    elif mychoice == 3:

        delete() #calling delete function to delete records

        db.commit()

        print()

        print()

        need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper()
#asking whether the user needs to continue or not

        if need_to_cont == "Y":

            main()

        elif need_to_cont == "N":

            db.close()

            print()

            print("program ended successfully")

            break

        else:

            db.close()

            print()

            print("invalid input")

            break

except Exception as e:

    print()

    print("OOPS! something's wrong") #error message

    print()

```

```

        print(e)
elif choice == 4:
    cursor.execute("SELECT * FROM studentstbl") #MySQL query command to display every
records of the studentstbl

    data = cursor.fetchall() # fetching data from the database

    print("student No : first name : last name")

    for item in data: #using for loop to print records one by one
        print(item)

    need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper() #asking
whether the user needs to continue or not

    print()
    print()
    if need_to_cont == "Y":
        main()
    elif need_to_cont == "N":
        db.close()
        print()
        print("program ended successfully")
        break
    else:
        db.close()
        print()
        print("invalid input")
        break

elif choice == 5:

```

```
cursor.execute("SELECT * FROM attendancetbl") #MySQL query command to display every records of the attendancetbl
```

```
data = cursor.fetchall()# fetching data from the database
```

```
print("student No : date : attendance")
```

```
print()
```

```
for item in data: #using for loop to print records one by one
```

```
    print(item)
```

```
print()
```

```
print()
```

```
need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper() #asking whether the user needs to continue or not
```

```
if need_to_cont == "Y":
```

```
    main()
```

```
elif need_to_cont == "N":
```

```
    db.close()
```

```
    print()
```

```
    print("program ended successfully")
```

```
    break
```

```
else:
```

```
    db.close()
```

```
    print()
```

```
    print("invalid input")
```

```
    break
```

```
elif choice == 6:
```

try: #exception handling

```
    stunum = input("enter the student number : ")
    print()
    print("student No : date : attendance")
    print()
    cursor.execute("SELECT * FROM attendancetbl WHERE studentNo = "+ stunum + "") #
MySQL query to show the attendance of the student, whose student number is entered above
    data = cursor.fetchall() # fetching data from the database
    print(data)
    print()
    print()
    need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper()
#asking whether the user needs to continue or not
    if need_to_cont == "Y":
        main()
    elif need_to_cont == "N":
        db.close()
        print()
        print("program ended successfully")
        break
    else:
        db.close()
        print()
        print("invalid input")
        break
except Exception as e:
    print()
```

```

    print("OOPS! something went wrong,please try again") #error message
    print()
    print(e)
    print()
    main()

elif choice == 7:
    print()
    print("left from the student attendance marker successfully!")
    break
    db.close() #disconnecting from MySQL server
else:
    print("please enter a valid input")
    need_to_cont = input("do you want to continue to main menu? (Y/N) : ").upper()
    if need_to_cont == "Y":
        main()
    elif need_to_cont == "N":
        db.close() #disconnecting from MySQL server
        print()
        print("program ended successfully")
        break
    else:
        db.close() #disconnecting from MySQL server
        print()
        print("invalid input")
        break

```