```
#*************************************************************************
#*************************************************************************
#
# Welcome to the book-issuing library program named "MyLib" written in
# Python.
# This program is developed by Madhurika Priya,Sweta Pandey and Anny
# Mondal.
#
# This program contains two files: MyLib1.py (main file) and Data1.py.
#
# This program requires the following to run properly:
# Python 3.0 or later, MySQL server and python3-MySQL connector.
#
#*************************************************************************
#*************************************************************************

#*************************************************************************
# Creates a connection to the MySQL server.
#
# Arguments of the mysql.connector.connect() function must be set
# properly:
# host: name of the host or server where mysql is located.
# user: name of the user who can access the mysql server.
# password: password of the user (if any).
#*************************************************************************

import mysql.connector
from datetime import date
import Data1    # Another python file of the program.

Cnx = mysql.connector.connect(
  host="localhost",
  user="root",
  password="12345"
)
# dictionary=True : Rows of the database table are stored as dictionary
# and not as tuple.
Cursor = Cnx.cursor(dictionary=True)

#*************************************************************************
# Function to print books in tabular format( Called by function
# 'ListBooks')
# Prints only necessary columns for better visibility.
#*************************************************************************
def PrintBooks(bookresult):
    columns = ('AccessNo', 'Name', 'Author', 'Issued to','Issue date')
    widths = [8,50,20,10,12]
    formatStr = '|'
    separator = '+'
    for w in widths:
        formatStr += " %-"+"%ss |" % (w,)
        separator += '-'*w + '--+'
    print(separator)
    print(formatStr % columns)
    print(separator)
    for row in bookresult:
        print(formatStr % (row['AccessNo'], row['Name'], row['Author'],
row['IssuedTo'], row['IssueDate']))
```

```
    print(separator)

#***************************************************************************
# Function to print a book detail in proper format (Called by function
# 'ListABookDetail')
#***************************************************************************
def PrintABookDetail(bookresult):
    Description = ('Name', 'Author', 'Type', 'Language', 'Year','Issued
to','Issue date')
    formatStr = "%-15s : "
    separator = '+'*100
    print(separator)
    print(formatStr % Description[0],bookresult['Name'])
    print(formatStr % Description[1],bookresult['Author'])
    print(formatStr % Description[2],bookresult['Type'])
    print(formatStr % Description[3],bookresult['Language'])
    print(formatStr % Description[4],bookresult['PubYear'])
    print(formatStr % Description[5],bookresult['IssuedTo'])
    print(formatStr % Description[6],bookresult['IssueDate'])
    print(separator)

#***************************************************************************
# Function to list all books
#***************************************************************************
def ListBooks():
    query = "select *from MyBooks"
    Cursor.execute(query)
    # fetchall returns a list of dictionary.
    bookresult = Cursor.fetchall()
    print("List of books in the library is: ")
    PrintBooks(bookresult)

#***************************************************************************
# This function lists detail of an individual book.
#***************************************************************************
def ListABookDetail():
    accessNo = int(input("enter Accession No.: "))
    query = "select *from MyBooks WHERE AccessNo = %s"
    vals = [(accessNo)]
    Cursor.execute(query,vals)
    # fetchone returns a dictionary.
    bookresult = Cursor.fetchone()
    if (Cursor.rowcount <= 0):
        print ("Sorry! Book having Accession no. %s not available in
library record." % accessNo)
        return
    print("Detail of the book having Accession no. %s is: " % accessNo)
    PrintABookDetail(bookresult)

#***************************************************************************
# Function to issue a book to a student
#***************************************************************************
def IssueABook():
    cmds = [
        '1=> by title',
        '2=> by author',
        '3=> by Accession No.'
    ]
```

```
    for j in cmds:
        print(j)
    inp = int(input("enter command: "))
    bookSought = ''
    if (inp == 1):
        bookSought = input("enter Book title: ")
        query = "select *from MyBooks where Name = '" + bookSought + "'"
    elif (inp == 2):
        bookSought = input("enter Book author: ")
        query = "select *from MyBooks where Author = '" + bookSought + "'"
    elif (inp == 3):
        bookSought = input("enter Book Accession No: ")
        query = "select *from MyBooks where AccessNo = '" + bookSought +
"'"

    Cursor.execute(query)
    bookresult = Cursor.fetchall()
    if (Cursor.rowcount <= 0):
        print("Sorry! Book having name/author as '%s' not available." %
bookSought)
        return
    print("List of books having the name/author as '%s' is : " %
bookSought)
    PrintBooks(bookresult)
    #
    # Cheack whether book sought is issued or not.
    nocopies = int(0)
    studentIssuedTo = ''
    for row in bookresult:
        if (row['IssuedTo'] == 0):
            nocopies = nocopies+1
        else:
            studentIssuedTo = row['IssuedTo']
    if (nocopies <= 0):
        print("Sorry! Book issued to :", studentIssuedTo)
        return
    print("Book Available! No. of copies is: ",nocopies)
    #
    # Ask for confirmation and Accession No. of the book.
    inp = input("Do you want to issue (y/n): ")
    if (inp != 'y'):
        return
    accessNo = int(input("enter Book Accession No.: "))
    query2 = "select *from MyBooks where AccessNo = " + str(accessNo) + "
AND IssuedTo= 0"
    Cursor.execute(query2)
    bookresult = Cursor.fetchone()
    if (Cursor.rowcount <= 0):
        print ("Sorry! Book (Accession no.: %s) is already issued or not
the book asked for issuing." % accessNo)
        return
    #
    # Enter roll no. of the student who sought the book.
    rollno = input("enter student roll no: ")
    query3 = "select *from Students where RollNo = " + rollno
    Cursor.execute(query3)
    studRes = Cursor.fetchone()
    if (Cursor.rowcount <= 0):
```

```
        print ("Sorry! student having roll no. %s not available in the
library record." % rollno)
        return
    #
    # Update the book with issuing record.
    query4 = "UPDATE MyBooks SET IssuedTo=%s,IssueDate=%s WHERE AccessNo =
%s"
    today_date = date.today()
    val = [(rollno,today_date,accessNo)]
    Cursor.executemany(query4,val)
    #
    # Print the sought book with updated record.
    Cursor.execute(query)
    bookresult = Cursor.fetchall()
    print("Updated list of books is: ")
    PrintBooks(bookresult)

#
*************************************************************************
# This function returns a book from a student.
#
*************************************************************************
def ReturnABook():
    # Ask for Accession No. of the book.
    accessNo = int(input("enter Accession No.: "))
    query = "select *from MyBooks WHERE AccessNo = %s"
    vals = [(accessNo)]
    Cursor.execute(query,vals)
    bookresult = Cursor.fetchone()
    if (Cursor.rowcount <= 0):
        print ("Sorry! Wrong Accession no. %s for the book." % accessNo)
        return
    #
    # Return the book by deleting IssuedTo record.
    query2 = "UPDATE MyBooks SET IssuedTo=%s,IssueDate=%s WHERE AccessNo =
%s"
    today_date = date.today()
    # set issue date as return date now
    vals2 = [(0,today_date,accessNo)]
    Cursor.executemany(query2,vals2)
    #
    # Print the returned book with updated record.
    Cursor.execute(query,vals)
    bookresult = Cursor.fetchone()
    print("Updated book detail having Accession no. %s is: " % accessNo)
    PrintABookDetail(bookresult)

#*************************************************************************
# Add New Arrival of book to the library
#*************************************************************************
def AddNewArrival():
    booktitle = input("enter Book title: ")
    bookauthor = input("enter Book author: ")
    accessNo = int(input("enter Accession No.: "))
    bookType = input("enter Book type: ")
    language = input("enter Language: ")
    pubYear = int(input("enter Publication Year: "))
    #
```

```
    # Check for the duplicate Accession No.
    query = "select *from MyBooks where AccessNo = " + str(accessNo)
    Cursor.execute(query)
    bookresult = Cursor.fetchone()
    if (Cursor.rowcount > 0):
        print ("Sorry! Book having Accession no. %s already exists." %
accessNo)
        return
    #
    # Insert new book into the database.
    query2 = "INSERT INTO MyBooks (AccessNo, Name, Author, Type, Language,
PubYear) VALUES (%s, %s, %s,%s,%s,%s)"
    val = [(accessNo, booktitle, bookauthor, bookType, language, pubYear)]
    Cursor.executemany(query2,val)
    #
    # Print the new book added.
    Cursor.execute(query)
    bookresult = Cursor.fetchone()
    print("New book detail having Accession no. %s is: " % accessNo)
    PrintABookDetail(bookresult)

#*************************************************************************
# Remove a book from the library record.
#*************************************************************************
def DeleteABook():
    # Ask for Accession No. of the book and check whether it exists.
    accessNo = int(input("enter Accession No.: "))
    query = "select *from MyBooks WHERE AccessNo = %s"
    vals = [(accessNo)]
    Cursor.execute(query,vals)
    bookresult = Cursor.fetchone()
    if (Cursor.rowcount <= 0):
        print ("Sorry! Wrong Accession no. %s for the book." % accessNo)
        return
    #
    # Delete the book.
    query = "DELETE FROM MyBooks WHERE AccessNo = %s"
    val = [(accessNo)]
    Cursor.execute(query,val)
    #
    # Print the updated list of books.
    query = "select *from MyBooks"
    Cursor.execute(query)
    bookresult = Cursor.fetchall()
    print("Updated list of books is: ")
    PrintBooks(bookresult)

#*************************************************************************
# Function to print students in tabular format
#*************************************************************************
def PrintStudents(studresult):
    print("List of students are: ")
    columns = ('RollNo', 'Name', 'Class', 'Section', 'DOB')
    widths = [10,40,6,8,12]
    formatStr = '|'
    separator = '+'
    for w in widths:
        formatStr += " %-"+"%ss |" % (w,)
```

```
        separator += '-'*w + '--+'
    print(separator)
    print(formatStr % columns)
    print(separator)
    for row in studresult:
        print(formatStr % (row['RollNo'], row['Name'], row['Class'],
row['Section'], row['DOB']))
    print(separator)

#**************************************************************************
# Function to print a student detail in proper format
#**************************************************************************
def PrintAStudentDetail(studresult):
    Description = ('Name', 'Class', 'Section', 'Date of Birth')
    formatStr = "%-20s : "
    separator = '+'*100
    print(separator)
    print(formatStr % Description[0],studresult['Name'])
    print(formatStr % Description[1],studresult['Class'])
    print(formatStr % Description[2],studresult['Section'])
    print(formatStr % Description[3],studresult['DOB'])
    print(separator)

#**************************************************************************
# Function to list all students
#**************************************************************************
def ListStudents():
    query = "select *from Students"
    Cursor.execute(query)
    studresult = Cursor.fetchall()
    PrintStudents(studresult)

#**************************************************************************
# Add new student
#**************************************************************************
def AddNewStudent():
    studentname = input("enter student name: ")
    rollNo = int(input("enter Roll No.: "))
    clss = input("enter class :")
    section = input("enter section: ")
    DOB = input("enter DOB (YYYY-MM-DD): ")
    #
    # Check for the duplicate Roll No.
    query = "select *from Students where RollNo = " + str(rollNo)
    Cursor.execute(query)
    studresult = Cursor.fetchone()
    if (Cursor.rowcount > 0):
        print ("Sorry! Student having Roll no. %s already exists." %
rollNo)
        return
    #
    # Insert new student into the database.
    query2 = "INSERT INTO Students (RollNo,Name,Class,Section,DOB) VALUES
(%s, %s, %s, %s, %s)"
    val = [(rollNo,studentname,clss,section,DOB)]
    Cursor.executemany(query2,val)
    #
    # Print the new student.
```

```
    Cursor.execute(query)
    studresult = Cursor.fetchone()
    print("New student detail having Roll no. %s is:" % rollNo)
    PrintAStudentDetail(studresult)

#************************************************************************
# Remove a student from the library record.
#************************************************************************
def DeleteAStudent():
    # Ask for Roll No. of the student and check whether it exists.
    rollNo = int(input("enter Roll No.: "))
    query = "select *from Students where RollNo = " + str(rollNo)
    Cursor.execute(query)
    studresult = Cursor.fetchone()
    if (Cursor.rowcount <= 0):
        print ("Sorry! Student having Roll no. %s doesn't exist." %
rollNo)
        return
    #
    # Delete the student.
    query2 = "DELETE FROM Students WHERE RollNo = %s"
    val = [(rollNo)]
    Cursor.execute(query2,val)
    #
    # Print the updated list of students.
    query = "select *from Students"
    Cursor.execute(query)
    studresult = Cursor.fetchall()
    print("Updated list of students is: ")
    PrintStudents(studresult)

#************************************************************************
# Update a student
#************************************************************************
def UpdateAStudent():
    # Ask for details of the student and check whether the student exists.

    rollNo = int(input("enter Roll No.: "))
    studentname = input("enter student name: ")
    clss = input("enter class :")
    section = input("enter section: ")
    DOB = input("enter DOB: ")
    query = "select *from Students where RollNo = " + str(rollNo)
    Cursor.execute(query)
    studresult = Cursor.fetchall()
    if (Cursor.rowcount <= 0):
        print ("Sorry! Wrong Roll no. %s for the student." % rollNo)
        return
    #
    # Update the student.
    query2 = "UPDATE Students SET Name=%s,Class=%s,Section=%s,DOB=%s WHERE
RollNo = %s"
    val = [(studentname,clss,section,DOB,rollNo)]
    Cursor.executemany(query2,val)
    #
    # Print the Updated student.
    Cursor.execute(query)
    studresult = Cursor.fetchone()
```

```
    print("Updated student detail having Roll no. %s is:" % rollNo)
    PrintAStudentDetail(studresult)

#************************************************************************
# Function to list books issued by a student
#************************************************************************
def ListBooksIssuedToAStudent():
    rollNo = int(input("enter Roll No.: "))
    query = "select *from MyBooks WHERE IssuedTo = %s"
    val = [(rollNo)]
    Cursor.execute(query,val)
    bookresult = Cursor.fetchall()
    if (Cursor.rowcount <= 0):
        print ("No book has been issued to the student (Roll no.: %s)." %
rollNo)
        return
    print("List of books issued to the student (Roll no. : %s) is: " %
rollNo)
    PrintBooks(bookresult)

#************************************************************************
# Main program
#************************************************************************

#************************************************************************
# Input command to use existing tables in the database or recreate all the
# tables in database.
#************************************************************************

cmds = [
    "***During installation, choose option '2' otherwise choose option
'1'.***",
    '1=> Use existing database, if exists',
    '2=> Create database'
]
for j in cmds:
    print(j)
inp = int(input("enter command: "))
if (inp == 1):
    Data1.UseExistingDatabase(Cursor)
elif (inp == 2):
    Data1.CreateDatabase(Cursor)

#************************************************************************
# Commands to use the library.
#************************************************************************

commands = [
    '1=> List books',
    '2=> Issue a book',
    '3=> Return a book',
    '4=> Add new arrival',
    '5=> Delete a book',
    '6=> List students',
    '7=> Add a new student',
    '8=> Update a student',
    '9=> Delete a student',
    '10=> List a book detail',
```

```
    '11=> List books issued by a student',
    '0=> Exit'
]
for j in commands:
    print(j)
inp = int(input("enter command: "))
while (inp != 0):
    if (inp == 1):
        ListBooks()
    elif (inp == 2):
        IssueABook()
    elif (inp == 3):
        ReturnABook()
    elif (inp == 4):
        AddNewArrival()
    elif (inp == 5):
        DeleteABook()
    elif (inp == 6):
        ListStudents()
    elif (inp == 7):
        AddNewStudent()
    elif (inp == 8):
        UpdateAStudent()
    elif (inp == 9):
        DeleteAStudent()
    elif (inp == 10):
        ListABookDetail()
    elif (inp == 11):
        ListBooksIssuedToAStudent()
    input("Press Enter to continue...")
    for j in commands:
        print(j)
    inp = int(input("enter command: "))
print("OK! Bye !")

#*************************************************************************
# Make sure data is committed (saved) to the database before exit.
#*************************************************************************
Cnx.commit()

Cursor.close()
Cnx.close()
#
# ******************************End of file*********************************
```