AMICUS INTERNATIONAL SCHOOL, BHARUCH

Practical File
of
Computer Science (083)

ACADEMIC YEAR: 2023-24

Submitted by:

Name:	Dipam Sen
Class:	XII (Science) - B
Roll No.:	

Certificate

This is to certify that **Dipam Sen**, student of Class XII, **Amicus International School, Bharuch** has completed the PRACTICAL FILE during the academic year **2023-24** towards partial fulfilment of credit for the Computer Science practical evaluation of CBSE and submitted a satisfactory report, as compiled in the following pages, under my supervision.

Signature of Internal Examiner	Signature of Principal	Signature of External Examiner

Acknowledgements

I extend my heartfelt gratitude to everyone who has contributed to the completion of this Project.

I am sincerely grateful to our principal Mrs. Thresiamma Pappachan and my Computer Science teacher Mrs. Arpita Bhoi. Their unwavering support and guidance have been instrumental in its completion.

Lastly, I express my sincere gratitude to my parents and family for their unwavering support.

To all those mentioned above and many others who have directly or indirectly contributed, your support has been truly invaluable in my growth as a learner.

- Dipam Sen

Index

Sr. No.	Program	Pg. No.	Date	Sign.
1	Write a user defined function to accept a string as an input and to count and display the total number of times a character is present in a string.	6	25/10/2023	
2	Write a program to compute the area of rectangle on the basis of length and breadth inputted by the user as the arguments to this function.	7	26/10/2023	
3	Write a menu driven program using different functions for the following menu: 1. Check no. is Palindrome or not 2. Check no. is Armstrong or not 3. Exit	8	26/10/2023	
4	Write a program using the function to print the Fibonacci series up to n numbers.	11	27/10/2023	
5	Write a random number generator using function that generates random numbers between 1 to 6 (simulates a dice).	12	27/10/2023	
6	Write a python program to read a file named "article.txt", count and print the following:	13	28/10/2023	
7	Read a text file and display the number of vowels/consonants/uppercase/lowercase characters in the file.	15	08/11/2023	
8	Read a text file line by line and display each word separated by a #.	17	12/11/2023	
9	Program to read and display those lines from the text file that starts with alphabet 'T'	18	13/11/2023	
10	Remove all the lines that contain the character 'a' in a file and write it to another file.	19	14/11/2023	

Sr. No.	Program	Pg. No.	Date	Sign.
11	Create a binary file with roll number and name. Search for a given roll number and display the name, if not found display appropriate message.	20	16/11/2023	
12	Create a binary file with the roll no. name and marks. Input a roll number and update the marks.	23	17/11/2023	
13	Write a program which adds any random five even numbers in a list that falls between the highest and lowest no. (Both highest the lowest numbers are accepted from the user)	25	20/11/2023	
14	Write a program using python to get 10 players name and their score and write it to a CSV file. Accept a player name using python and search it in the CSV file.	26	21/11/2023	
15	Create a CSV file by entering user-id and password, read and search the password for given user-id.	29	04/12/2023	
16	Write a python program using function `PUSH(Arr)`, where `Arr` is a list of numbers.	31	21/11/2023	
17	Write a python program using function `POP(Arr)`, where `Arr` is a stack implemented by a list of numbers.	32	22/11/2023	
18	Write a python program to integrate MySQL with Python by inserting records to `EMP` table and displaying records.	34	22/11/2023	
19	Create an Employee Table with the fields `Empno`, `Empname`, `Desig`, `Dept`, `Age` and `Place`. Enter five records into the table.	37	23/11/2023	
20	Create Student table with following fields and enter data as given in the table below. Then query the following.	39	24/11/2023	

Program 1

Write a user defined function to accept a string as an input and to count and display the total number of times a character is present in a string.

Program:

```python
def count_occ(str, ch):
    count = 0
    for i in str:
        if i == ch:
            count += 1
    return count


val = input("Enter a string: ")
c = input("Enter a character: ")
num = count_occ(val, c)

print("The character occurs " + str(num) +
      " times in the string.")
```

Output:

```
Enter a string: Computer Science
Enter a character: e
The character occurs 3 times in the string.
```

Program 2

Write a program to compute the area of rectangle on the basis of length and breadth inputted by the user as the arguments to this function.

Program:

```
def area(l, b):
  return l * b


length = int(input("Enter length: "))
breadth = int(input("Enter breadth: "))

val = area(length, breadth)

print("Area of the rectangle is", val)
```

Output:

```
Enter length: 20
Enter breadth: 40
Area of the rectangle is 800
```

Program 3

Write a menu driven program using different functions for the following menu:

1. Check no. is Palindrome or not
2. Check no. is Armstrong or not
3. Exit

Program:

```
def is_palindrome(num):
  s = str(num)
  if s == s[::-1]:
    return True
  return False


def is_armstrong(num):
  n = len(str(num))
  total = 0
  for digit in str(num):
    total += int(digit)**n
  if total == num:
    return True
  return False


while True:
  print("=======================")
  print("Menu")
  print("======================")
  print()
  print("1. Check if number is Palindrome")
  print("2. Check if number is Armstrong")
  print("3. Exit")
  choice = int(input("Enter your choice (1-3): 
"))
  if choice == 1:
    num = int(input("Enter your number: "))
    if is_palindrome(num):
      print("Your number is a palindrome!")
    else:
      print("Your number is not a palindrome!")
  elif choice == 2:
    num = int(input("Enter your number: "))
    if is_armstrong(num):
      print("Your number is armstrong!")
    else:
      print("Your number is not angstrom!")
  elif choice == 3:
    break
  else:
    continue
```

Output:

```
======================
Menu
======================
1. Check if number is Palindrome
2. Check if number is Armstrong
3. Exit
Enter your choice (1-3): 1
Enter your number: 21412
Your number is a palindrome!
======================
Menu
======================

1. Check if number is Palindrome
2. Check if number is Armstrong
3. Exit
Enter your choice (1-3): 2
Enter your number: 1634
Your number is armstrong!
======================
Menu
======================

1. Check if number is Palindrome
2. Check if number is Armstrong
3. Exit
Enter your choice (1-3): 3
```

Program 4

Write a program using the function to print the Fibonacci series up to n numbers.

Program:

```
def fibonacci(n):
    series = [0, 1]
    while len(series) < n:
        a = series[-1]
        b = series[-2]
        # next term = sum of last two terms
        series.append(a + b)
    return series


num = int(
    input("Enter length of Fibonacci Sequence:
"))
vals = fibonacci(num)
for val in vals:
    print(val, end="\t")
```

Output:

```
Enter length of Fibonacci Sequence: 12
0       1       1       2       3       5
8       13      21      34      55      89
```

Program 5

Write a random number generator using function that generates random numbers between 1 to 6 (simulates a dice).

Program:

```
import random


def dice():
    return random.randint(1, 6)


print("🎲 Rolling the dice 🎲")
print("You got", dice(), "!")
```

Output:

```
🎲 Rolling the dice 🎲
You got 1 !
```

Program 6

Write a python program to read a file named “article.txt”, count and print the following:

(i) total alphabets
(ii) total upper case alphabets
(iii) total lower case alphabets
(iv) total digits
(v) total spaces
(vi) total special characters

Program:

```python
alpha = 0
upper = 0
lower = 0
digit = 0
space = 0
spchr = 0

f = open("./article.txt")
data = f.read()
for ch in data:
    if ch.isalpha():
        alpha += 1
    if ch.isupper():
        upper += 1
    if ch.islower():
        lower += 1
    if ch.isdigit():
            digit += 1
        if ch.isspace():
            space += 1
        if not ch.isalnum() and not ch.isspace():
            spchr += 1

print("Total alphabets:", alpha)
print("Total uppercase:", upper)
print("Total lowercase:", lower)
print("Total digits:", digit)
print("Total spaces:", space)
print("Total special characters:", spchr)

f.close()
```

article.txt

```
Hello World!

Sample text 0123456789 #$%^&*()
```

Output:

```
Total alphabets: 20
Total uppercase: 3
Total lowercase: 17
Total digits: 10
Total spaces: 7
Total special characters: 9
```

Program 7

Read a text file and display the number of vowels/ consonants/uppercase/lowercase characters in the file.

Program:

```
upper = 0
lower = 0
vowel = 0
conso = 0

vowellist = "aeiou"

with open("./article.txt") as f:
    data = f.read()
    for char in data:
        if char.isupper():
            upper += 1
        if char.islower():
            lower += 1
        if char.isalpha():
            if char.lower() in vowellist:
                vowel += 1
            else:
                conso += 1

    print("Total vowels:", vowel)
    print("Total consonants:", conso)
    print("Total uppercase:", upper)
    print("Total lowercase:", lower)
```

article.txt

```
Python
SQL
File Handling
```

Output:

```
Total vowels: 5
Total consonants: 16
Total uppercase: 6
Total lowercase: 15
```

Program 8

Read a text file line by line and display each word separated by a #.

Program:

```
f = open("article.txt")
l = " "
while l:
    l = f.readline()
    words = l.split(" ")
    print("#".join(words), end="")
f.close()
```

article.txt

```
Types of functions
Creating user defined function
Arguments and Parameters
Function returning value
```

Output:

```
Types#of#functions
Creating#user#defined#function
Arguments#and#Parameters
Function#returning#value
```

Program 9

Program to read and display those lines from the text file that starts with alphabet ‘T’

Program:

```
f = open("article.txt")
l = ' '

while l:
    l = f.readline()
    if l.startswith("T"):
        print(l.strip())
```

article.txt

```
Time
Random
Tkinter
Numpy
Math
Turtle
```

Output:

```
Time
Tkinter
Turtle
```

Program 10

Remove all the lines that contain the character ‘a’ in a file and write it to another file.

Program:

```
file1 = open("article.txt")
file2 = open("output.txt", "w")

l = " "
while l:
    l = file1.readline()
    if "a" not in l:
        file2.write(l)

file2.close()
```

`article.txt`

```
Computer Systems and Organisation
Society & Ethics
Computational Thinking and Programming
Computer Networks
```

`output.txt` (Output)

```
Society & Ethics
Computer Networks
```

Program 11

Create a binary file with roll number and name. Search for a given roll number and display the name, if not found display appropriate message.

Program:

Data Entry

```python
import pickle

print("Students Data Entry")
print()

file = open("students.dat", "wb")
count = 0
run = True
while run:
    name = input("Enter name: ")
    roll = int(input("Enter roll no: "))
    obj = {"name": name, "roll": roll}
    pickle.dump(obj, file)
    count += 1
    print()
    run = input("Add more? (y/n) ") == "y"

print(f"Successfully entered {count}"
      " entries to students.dat.")
file.close()
```

Main Code

```
import pickle

roll = int(input("Enter roll number to search:
"))
found = False
try:
    file = open("students.dat", "rb")
    while True:
        stu = pickle.load(file)
        if stu["roll"] == roll:
            print("Found:")
            print(stu["name"],
                  stu["roll"],
                  sep="\t")
            found = True
            break
except EOFError:
    if not found:
        print(f"Could not find any entries"
              " for roll no. {roll}.")
    file.close()
except FileNotFoundError:
    print("Could not find binary file.")
```

Output:

```
Students Data Entry

Enter name: Anusha
Enter roll no: 1

Add more? (y/n) y
Enter name: Dhruv
Enter roll no: 2

Add more? (y/n) y
Enter name: Ramesh
Enter roll no: 3

Add more? (y/n) y
Enter name: Shyam
Enter roll no: 4

Add more? (y/n) n
Successfully entered 4 entries to students.dat.
```

```
Enter roll number to search: 3
Found:
Ramesh  3
```

Program 12

Create a binary file with the roll no. name and marks. Input a roll number and update the marks.

Program:

Data Entry — *Similar to program 11*

Main Code

```
import pickle

data = []
# data: {"name": "", "roll": 0, "mark": 0}
file = open("students.dat", "rb")
rno = int(input("Enter roll no to update: "))
found = False

while True:
    try:
        stu = pickle.load(file)
        if stu["roll"] == rno:
            found = True
            print("Found entry: ")
            print(stu["roll"],
                  stu["name"],
                  stu["mark"],
                  sep="\t")
            print()
            marks = int(
                input("Enter new marks: "))
            stu["mark"] = marks
        data.append(stu)
    except EOFError:
        break

if not found:
    print("No entry found for roll no", rno)
else:
    fw = open("students.dat", "wb")
    for stu in data:
        pickle.dump(stu, fw)
    print("Successfully updated students.dat.")
    fw.close()

file.close()
```

Output:

```
Enter roll no to update: 3
Found entry:
3     Ramesh    83

Enter new marks: 86
Successfully updated students.dat.
```

Program 13

Write a program which adds any random five even numbers in a list that falls between the highest and lowest no. (Both highest the lowest numbers are accepted from the user)

Program:

```
import random

lo = int(input("Enter lower bound: "))
hi = int(input("Enter upper bound: "))

lst = []

while len(lst) < 5:
  num = random.randint(lo + 1, hi - 1)
  if num % 2 == 0:
    lst.append(num)

print("The list of numbers is: ", end="")
print(lst)
```

Output:

```
Enter lower bound: 10
Enter upper bound: 32
The list of numbers is: [26, 28, 22, 30, 12]
```

Program 14

Write a program using python to get 10 players name and their score. Write the input in a CSV file. Accept a player name using python. Read the CSV file to display the name and the score. If the player name is not found give an appropriate message.

Program:

```
import csv

f = open("players.csv", "w", newline="")
writer = csv.writer(f)
writer.writerow(["name", "score"])
for i in range(10):
    print("Player", i + 1)
    name = input("Enter Name: ")
    score = int(input("Enter Score: "))
    writer.writerow([name, score])

f.close()

# -------------------------

print('\n' * 3)
print("Player Search")
search = input("Enter name: ")
file = open("players.csv", newline="")
rdr = csv.reader(file)
```

```
for player in rdr:
    name, score = player
    if name == search:
        print("Found player")
        print(f"{name} - Score: {score}")
        break
else:
    print("Did not find any player matching",
          search)
```

Output:

```
Player 1
Enter Name: Parth
Enter Score: 20
Player 2
Enter Name: Harsh
Enter Score: 23
Player 3
Enter Name: Parth
Enter Score: 29
Player 4
Enter Name: Kapil
Enter Score: 25
Player 5
Enter Name: Sparsh
Enter Score: 23
Player 6
Enter Name: Soham
Enter Score: 18
Player 7
Enter Name: Pranjal
Enter Score: 23
Player 8
Enter Name: Rohan
Enter Score: 16
Player 9
Enter Name: Dev
Enter Score: 20
Player 10
Enter Name: Kushal
Enter Score: 13




Player Search
Enter name: Pranjal
Found player
Pranjal - Score: 23
```

Program 15

Create a CSV file by entering user-id and password, read and search the password for given user-id.

Program:

```
import csv

f = open("data.csv", "w", newline="")
writer = csv.writer(f)
writer.writerow(["userid", "password"])
while True:
    id = input("Enter User ID: ")
    pwd = input("Enter Password: ")
    writer.writerow([id, pwd])

    if input("More entries? (y/n) ") != "y":
        break

f.close()

# -------------------------

print('\n' * 2)
search = input("Enter User ID: ")
file = open("data.csv", newline="")
rdr = csv.reader(file)
for user in rdr:
    id, pwd = user
    if id == search:
        print(f"ID: {id}\t Password: {pwd}")
        break
else:
    print("Did not find any user matching",
          search)
```

Output:

```
Enter User ID: python67
Enter Password: password123
More entries? (y/n) y
Enter User ID: patel23
Enter Password: admin
More entries? (y/n) y
Enter User ID: soham
Enter Password: @123456
More entries? (y/n) n



Enter User ID: patel23
ID: patel23      Password: admin
```

Program 16

Write a python program using function `PUSH(Arr)`, where `Arr` is a list of numbers. From this list push all numbers divisible by 5 into a stack implemented by using a list. Display the stack if it has at least one element, otherwise display appropriate error message.

Program:

```
# Arr -> stack implemented by list
def PUSH(Arr, elt):
    Arr.append(elt)


Arr = [23, 30, 34, 91, 29, 48, 35, 90, 27]
stack = []
for num in Arr:
    if num % 5 == 0:
        PUSH(stack, num)
if len(stack) > 0:
    print(stack)
else:
    print("Stack is empty!")
```

Output:

```
[30, 35, 90]
```

Program 17

Write a python program using function `POP(Arr)`, where `Arr` is a stack implemented by a list of numbers. The function returns the value deleted from the stack.

Program:

```python
# Arr -> stack implemented by list
def POP(Arr):
    if len(Arr) == 0:
        raise Exception("Underflow")
    item = Arr.pop()
    return item


Arr = [20, 40, 50, 60]
print("Initial stack:", Arr)
print("Started popping...")
try:
    while True:
        val = POP(Arr)
        print("Got", val)
except Exception:
    print("Underflow")
```

Output:

```
Initial stack: [20, 40, 50, 60]
Started popping...
Got 60
Got 50
Got 40
Got 20
Underflow
```

Program 18

Write a python program to integrate MySQL with Python by inserting records to EMP table and displaying records.

Program:

```
import mysql.connector as sql

conn = sql.connect(user="root",
                   host="localhost",
                   passwd="password",
                   database="test")
curr = conn.cursor()


def insert(ID, name, age, dept):
    curr.execute(
        """INSERT INTO Emp (id, name, age, dept)
           VALUES (%s, %s, %s, %s)""",
        (ID, name, age, dept))
    conn.commit()


def display():
    curr.execute("SELECT * FROM Emp")
    print("ID\tName\t\tDept\tAge")
    print("=" * 40)
    data = curr.fetchall()
    for emp in data:
        print(*emp, sep="\t")
```

```python
while True:
    print()
    print("(1) Show employees",
          "(2) Insert employee", "(3) Exit")
    choice = int(
        input("Enter your choice: (1/2/3) "))
    if choice == 1:
        display()
    elif choice == 2:
        name = input("Name: ")
        ID = int(input("ID: "))
        dept = input("Dept: ")
        age = int(input("Age: "))
        insert(ID, name, age, dept)
        print("Successfully inserted employee.")
    elif choice == 3:
        break
```

Output:

```
(1) Show employees (2) Insert employee (3) Exit
Enter your choice: (1/2/3) 1
ID      Name            Dept    Age
========================================
1       Sameer Sharma   Acc     39
2       Rama Gupta      HR      43
3       C R Menon       IT      48
4       Sanjeev P       Res     54
```

```
(1) Show employees (2) Insert employee (3) Exit
Enter your choice: (1/2/3) 2
Name: Rajesh Kumar
ID: 5
Dept: IT
Age: 42
Successfully inserted employee.

(1) Show employees (2) Insert employee (3) Exit
Enter your choice: (1/2/3) 1
ID      Name            Dept    Age
========================================
1       Sameer Sharma   Acc     39
2       Rama Gupta      HR      43
3       C R Menon       IT      48
4       Sanjeev P       Res     54
5       Rajesh Kumar    IT      42

(1) Show employees (2) Insert employee (3) Exit
Enter your choice: (1/2/3) 3
```

Program 19

Create an Employee Table with the fields `Empno`, `Empname`, `Desig`, `Dept`, `Age` and `Place`. Enter five records into the table.

- Add two more records to the table.
- Modify the table structure by adding one more field namely date of joining. (`doj`)
- Check for `NULL` value in `doj` of any record.

Code and Output:

```
CREATE TABLE Employee (
    Empno INT PRIMARY KEY,
    Empname VARCHAR(255),
    Desig VARCHAR(50),
    Dept VARCHAR(10),
    Age INT,
    Place VARCHAR(50)
);

INSERT INTO Employee VALUES
    (1, "Rajesh Kumar", "General Manager",
"HRD", 42, "Hyderabad"),
    (2, "Sameer Sharma", "Manager", "IT", 38,
"Bhopal"),
    (3, "C R Menon", "Senior Manager",
"Quality", 36, "Chennai"),
    (4, "Mahesh Arora", "Assistant Manager",
"Research", 45, "Bangalore"),
    (5, "Ramesh Murthy", "CPO", "IT", 44,
"Ahmedabad");
```

```
-- Add two more records to the table.
> INSERT INTO Employee
    (Empno, Empname, Desig, Dept, Age, Place)
    VALUES
    (6, "Abdul Ahmed", "CEO", "Quality", 47,
"Aurangabad"),
    (7, "Priyam Sen", "Manager", "HRD", 41,
"Kolkata");

Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
-- Modify the table structure by adding one more
field namely date of joining.
> ALTER TABLE Employee ADD doj DATE;

Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
-- Check for Null value in doj of any record.
> SELECT * FROM Employee WHERE doj IS NULL;

+-------+---------------+-------------------+----------+------+------------+------+
| Empno | Empname       | Desig             | Dept     | Age  | Place      | doj  |
+-------+---------------+-------------------+----------+------+------------+------+
|     1 | Rajesh Kumar  | General Manager   | HRD      |   42 | Hyderabad  | NULL |
|     2 | Sameer Sharma | Manager           | IT       |   38 | Bhopal     | NULL |
|     3 | C R Menon     | Senior Manager    | Quality  |   36 | Chennai    | NULL |
|     4 | Mahesh Arora  | Assistant Manager | Research |   45 | Bangalore  | NULL |
|     5 | Ramesh Murthy | CPO               | IT       |   44 | Ahmedabad  | NULL |
|     6 | Abdul Ahmed   | CEO               | Quality  |   47 | Aurangabad | NULL |
|     7 | Priyam Sen    | Manager           | HRD      |   41 | Kolkata    | NULL |
+-------+---------------+-------------------+----------+------+------------+------+
7 rows in set (0.00 sec)
```

Program 20

Create Student table with following fields and enter data as given in the table below.

Field	Type	Size
Reg_No	char	5
Sname	varchar	15
Age	int	2
Dept	varchar	10
Class	char	3

Reg_No	Sname	Age	Dept	Class
M1001	Harish	19	ME	ME1
M1002	Akash	20	ME	ME2
C1001	Sneha	20	CSE	CS1
C1002	Lithya	19	CSE	CS2
E1001	Ravi	20	ECE	EC1
E1002	Leena	21	EEE	EE1
E1003	Rose	20	ECE	EC2

Then, query the following:

(i) List the students whose department is “CSE”.

(ii) List all the students of age 20 and more in ME department.

(iii) List the students department wise.

(iv) Modify the class ME2 to ME1.

Code and Output:

```
CREATE TABLE Student (
    Reg_No CHAR(5),
    Sname VARCHAR(15),
    Age INT(2),
    Dept VARCHAR(10),
    Class CHAR(3)
);
```

```
INSERT INTO Student VALUES
    ("M1001", "Harish", 19, "ME", "ME1"),
    ("M1002", "Akaash", 20, "ME", "ME2"),
    ("C1001", "Sneha", 20, "CSE", "CS1"),
    ("C1002", "Lithya", 19, "CSE", "CS2"),
    ("E1001", "Ravi", 20, "ECE", "EC1"),
    ("E1002", "Leena", 21, "EEE", "EE1"),
    ("E1003", "Rose", 20, "ECE", "EC2");
```

```
-- List the students whose department is “CSE”.
> SELECT * FROM Student WHERE Dept = "CSE";

+--------+--------+------+------+-------+
| Reg_No | Sname  | Age  | Dept | Class |
+--------+--------+------+------+-------+
| C1001  | Sneha  |   20 | CSE  | CS1   |
| C1002  | Lithya |   19 | CSE  | CS2   |
+--------+--------+------+------+-------+
2 rows in set (0.00 sec)
```

```
-- List all the students of age 20 and more in
ME department.
> SELECT * FROM Student
    WHERE Age >= 20
    AND Dept = "ME";

+--------+--------+------+------+-------+
| Reg_No | Sname  | Age  | Dept | Class |
+--------+--------+------+------+-------+
| M1002  | Akaash |   20 | ME   | ME2   |
+--------+--------+------+------+-------+
1 row in set (0.00 sec)
```

```
--  List the students department wise.
> SELECT Dept, COUNT(*) FROM Student
    GROUP BY Dept;

+------+----------+
| Dept | COUNT(*) |
+------+----------+
| ME   |        2 |
| CSE  |        2 |
| ECE  |        2 |
| EEE  |        1 |
+------+----------+
4 rows in set (0.00 sec)
```

```
-- Modify the class ME2 to ME1.
> UPDATE Student SET Class = "ME1"
    WHERE Class = "ME2";

Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

> SELECT * FROM Student;

+--------+--------+------+------+-------+
| Reg_No | Sname  | Age  | Dept | Class |
+--------+--------+------+------+-------+
| M1001  | Harish |   19 | ME   | ME1   |
| M1002  | Akaash |   20 | ME   | ME1   |
| C1001  | Sneha  |   20 | CSE  | CS1   |
| C1002  | Lithya |   19 | CSE  | CS2   |
| E1001  | Ravi   |   20 | ECE  | EC1   |
| E1002  | Leena  |   21 | EEE  | EE1   |
| E1003  | Rose   |   20 | ECE  | EC2   |
+--------+--------+------+------+-------+
7 rows in set (0.00 sec)
```

Bibliography

1. Sumita Arora: Computer Science with Python — Textbook for Class XII. Dhanpat Rai and Co. (2023)
2. The Python Standard Library Documentation, https://docs.python.org/3/library/index.html
3. io — Core tools for working with streams, https://docs.python.org/3/library/io.html
4. csv — CSV File Reading and Writing, https://docs.python.org/3/library/csv.html
5. MySQL 8.2 Reference, https://dev.mysql.com/doc/refman/8.2/en/