SET 2

Practical 1

Ques 1: Write a program in python that models a bank account, with classes for the account, the customer, and the bank.

Aim: modelling a bank account system

CODE:

```
class Customer:
    def __init__(self, customer_id, name):
        self.customer_id = customer_id
        self.name = name

    def __str__(self):
        return f"Customer ID: {self.customer_id}, Name: {self.name}"

class Account:
    def __init__(self, account_number, customer, balance=0):
        self.account_number = account_number
        self.customer = customer
        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited {amount}. New balance: {self.balance}")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount):
        if amount > 0 and amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew {amount}. New balance: {self.balance}")
        elif amount > self.balance:
            print("Insufficient balance.")
        else:
            print("Withdrawal amount must be positive.")

    def __str__(self):
        return f"Account Number: {self.account_number}, Customer: {self.customer.name},
Balance: {self.balance}"

class Bank:
    def __init__(self, name):
        self.name = name
        self.customers = []
        self.accounts = []

    def add_customer(self):
        customer_id = input("Enter Customer ID: ")
        name = input("Enter Customer Name: ")
        customer = Customer(customer_id, name)
        self.customers.append(customer)
        print("Customer added successfully!\n")

    def open_account(self):
        if not self.customers:
            print("No customers available. Please add a customer first.")
            return

        print("Available Customers:")
        for idx, customer in enumerate(self.customers):
            print(f"{idx + 1}. {customer.name}")

        customer_choice = int(input("Select a customer by number: ")) - 1
        if 0 <= customer_choice < len(self.customers):
            account_number = input("Enter Account Number: ")
            initial_deposit = float(input("Enter Initial Deposit Amount: "))
            account = Account(account_number, self.customers[customer_choice],
  initial_deposit)
            self.accounts.append(account)
            print("Account opened successfully!\n")
        else:
            print("Invalid choice. Please try again.\n")

    def view_accounts(self):
        if not self.accounts:
            print("No accounts to display.\n")
            return
        print("Accounts:")
        for account in self.accounts:
            print(account)
        print()

    def perform_transaction(self):
        if not self.accounts:
            print("No accounts available. Please open an account first.")
            return

        account_number = input("Enter Account Number: ")
        account = next((acc for acc in self.accounts if acc.account_number ==
account_number), None)

        if not account:
            print("Account not found.")
            return

        print("1. Deposit")
        print("2. Withdraw")
        choice = input("Enter your choice: ")

        if choice == "1":
            amount = float(input("Enter amount to deposit: "))
            account.deposit(amount)
        elif choice == "2":
            amount = float(input("Enter amount to withdraw: "))
            account.withdraw(amount)
        else:
            print("Invalid choice.")

    def run(self):
        while True:
            print("\nBank Management System")
            print("1. Add Customer")
            print("2. Open Account")
            print("3. View Accounts")
            print("4. Perform Transaction")
            print("5. Exit")
            choice = input("Enter your choice: ")

            if choice == "1":
                self.add_customer()
            elif choice == "2":
                self.open_account()
            elif choice == "3":
                self.view_accounts()
            elif choice == "4":
                self.perform_transaction()
            elif choice == "5":
                print("Exiting the system. Goodbye!")
                break
            else:
                print("Invalid choice. Please try again.\n")

# Running the program
if __name__ == "__main__":
    bank = Bank("My Bank")
    bank.run()
```

OUTPUT
**you can write your own output

```
Bank Management System
1. Add Customer
2. Open Account
3. View Accounts
4. Perform Transaction
5. Exit
Enter your choice: >? 1
Enter Customer ID: >? 123
Enter Customer Name: >? anjali
Customer added successfully!
Bank Management System
1. Add Customer
2. Open Account
3. View Accounts
4. Perform Transaction
5. Exit
Enter your choice: >? 2
Available Customers:
1. anjali
Select a customer by number: >? 1
Enter Account Number: >? 123456789
Enter Initial Deposit Amount: >? 24000
Account opened successfully!
Bank Management System
1. Add Customer
2. Open Account
3. View Accounts
4. Perform Transaction
5. Exit
Enter your choice: >? 3
Accounts:
Account Number: 123456789, Customer: anjali, Balance: 24000.0
Bank Management System
1. Add Customer
2. Open Account
3. View Accounts
4. Perform Transaction
5. Exit
Enter your choice: >? 4
Enter Account Number: >? 123456789
1. Deposit
2. Withdraw
Enter your choice: >? 1
Enter amount to deposit: >? 2300
Deposited 2300.0. New balance: 26300.0
```

```
Bank Management System
1. Add Customer
2. Open Account
3. View Accounts
4. Perform Transaction
5. Exit
Enter your choice: >? 5
Exiting the system. Goodbye!
```

Practical 2:

QUES 2: Write a program in python that simulates a school management system, with classes for the students, the teachers, and the courses.

Aim: to manage school related work

Code:

```python
class Student:
    def __init__(self, student_id, name, age):
        self.student_id = student_id
        self.name = name
        self.age = age

    def __str__(self):
        return f"ID: {self.student_id}, Name: {self.name}, Age: {self.age}"

class Teacher:
    def __init__(self, teacher_id, name, subject):
        self.teacher_id = teacher_id
        self.name = name
        self.subject = subject

    def __str__(self):
        return f"ID: {self.teacher_id}, Name: {self.name}, Subject: {self.subject}"

class Course:
    def __init__(self, course_id, name, teacher):
        self.course_id = course_id
        self.name = name
        self.teacher = teacher

    def __str__(self):
        return f"Course ID: {self.course_id}, Name: {self.name}, Teacher: {self.teacher.name}"
```

```
class SchoolManagementSystem:
    def __init__(self):
        self.students = []
        self.teachers = []
        self.courses = []

    def add_student(self):
        student_id = input("Enter Student ID: ")
        name = input("Enter Student Name: ")
        age = input("Enter Student Age: ")
        student = Student(student_id, name, age)
        self.students.append(student)
        print("Student added successfully!\n")

    def add_teacher(self):
        teacher_id = input("Enter Teacher ID: ")
        name = input("Enter Teacher Name: ")
        subject = input("Enter Subject Taught by Teacher: ")
        teacher = Teacher(teacher_id, name, subject)
        self.teachers.append(teacher)
        print("Teacher added successfully!\n")

    def add_course(self):
        course_id = input("Enter Course ID: ")
        name = input("Enter Course Name: ")
        if not self.teachers:
            print("No teachers available. Please add a teacher first.")
            return
        print("Available Teachers:")
        for idx, teacher in enumerate(self.teachers):
            print(f"{idx + 1}. {teacher.name} - {teacher.subject}")
        teacher_choice = int(input("Select a teacher by number: ")) - 1
        if 0 <= teacher_choice < len(self.teachers):
            course = Course(course_id, name, self.teachers[teacher_choice])
            self.courses.append(course)
            print("Course added successfully!\n")
        else:
            print("Invalid choice. Please try again.\n")

    def view_students(self):
        if not self.students:
            print("No students to display.\n")
            return
        print("Students:")
        for student in self.students:
            print(student)
        print()
```

```
def view_teachers(self):
    if not self.teachers:
        print("No teachers to display.\n")
        return
    print("Teachers:")
    for teacher in self.teachers:
        print(teacher)
    print()

def view_courses(self):
    if not self.courses:
        print("No courses to display.\n")
        return
    print("Courses:")
    for course in self.courses:
        print(course)
    print()

def run(self):
    while True:
        print("School Management System")
        print("1. Add Student")
        print("2. Add Teacher")
        print("3. Add Course")
        print("4. View Students")
        print("5. View Teachers")
        print("6. View Courses")
        print("7. Exit")
        choice = input("Enter your choice: ")

        if choice == "1":
            self.add_student()
        elif choice == "2":
            self.add_teacher()
        elif choice == "3":
            self.add_course()
        elif choice == "4":
            self.view_students()
        elif choice == "5":
            self.view_teachers()
        elif choice == "6":
            self.view_courses()
        elif choice == "7":
            print("Exiting the system. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.\n")

# Running the program
if __name__ == "__main__":
    system = SchoolManagementSystem()
    system.run()
```

OUTPUT

**you can write your own output

```
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice: >? 1
Enter Student ID: >? 123
Enter Student Name: >? anjali
Enter Student Age: >? 22
Student added successfully!
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice: >? 3
Enter Course ID: >? 12
Enter Course Name: >? python
No teachers available. Please add a teacher first.
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice: >? 2
Enter Teacher ID: >? 345
Enter Teacher Name: >? manya
Enter Subject Taught by Teacher: >? python
Teacher added successfully!
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice: >? 4
Students:
ID: 123, Name: anjali, Age: 22
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice: >? 5
Teachers:
ID: 345, Name: manya, Subject: python
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice: >? 3
Enter Course ID: >? 12
Enter Course Name: >? python
Available Teachers:
1. manya - python
Select a teacher by number: >? 1
Course added successfully!
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
```

```
Enter your choice: >? 6
Courses:
Course ID: 12, Name: python, Teacher: manya
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice: >? 7
Exiting the system. Goodbye!
```

Practical 3:

QUES 3: Write a program in python that reads a text file and counts the number of words in it.

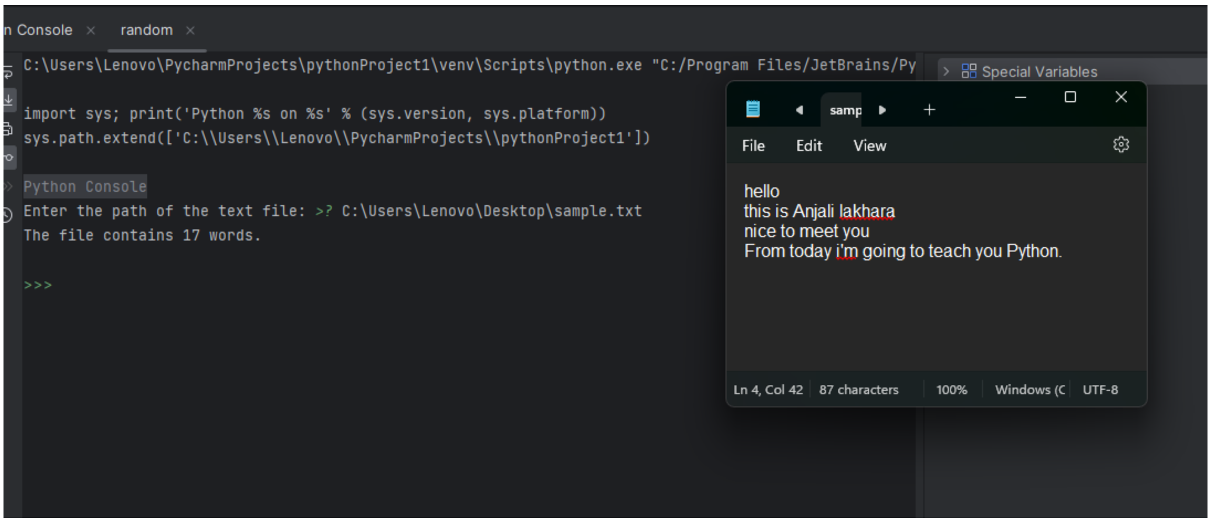
Aim: to read and count all the characters in your file

Code:

```
def count_words_in_file():
    file_path = input("Enter the path of the text file: ")

    try:
        with open(file_path, 'r') as file:
            text = file.read()
            words = text.split()
            word_count = len(words)
            print(f"The file contains {word_count} words.")
    except FileNotFoundError:
        print("File not found. Please check the file path and try again.")
    except Exception as e:
        print(f"An error occurred: {e}")

if __name__ == "__main__":
    count_words_in_file()
```

OUTPUT

First create a notepad file.



```
C:\Users\Lenovo\PycharmProjects\pythonProject1\venv\Scripts\python.exe "C:/Program Files/JetBrains/Py

import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\Users\\Lenovo\\PycharmProjects\\pythonProject1'])

Python Console
Enter the path of the text file: >? C:\Users\Lenovo\Desktop\sample.txt
The file contains 17 words.

>>>
```

```
hello
this is Anjali lakhara
nice to meet you
From today i'm going to teach you Python.
```

Ln 4, Col 42 | 87 characters | 100% | Windows (C | UTF-8

Practical 4:

QUES 4: Write a program in python that reads a CSV file and calculates the average of the values in a specified column.

Aim: reading a **Comma Separated Value** file in python

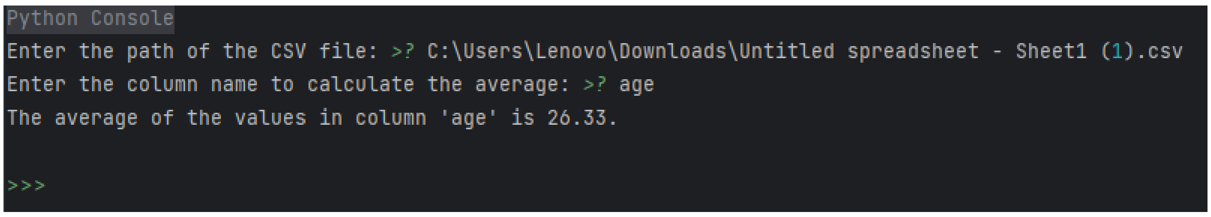
Code:

```
import csv

def calculate_column_average():
    file_path = input("Enter the path of the CSV file: ")
    column_name = input("Enter the column name to calculate the average: ")

    try:
        # Open the CSV file and read it
        with open(file_path, 'r') as csvfile:
            reader = csv.DictReader(csvfile)
            values = []

            for row in reader:
                try:
                    # Try to convert the value to a float
                    value = float(row[column_name])
                    values.append(value)
                except KeyError:
```

```
                print(f"Column '{column_name}' not found in the CSV file.")
                return
            except ValueError:
                print(f"Non-numeric value encountered in column '{column_name}'.
Skipping...")

        if values:
            # Calculate the average
            average = sum(values) / len(values)
            print(f"The average of the values in column '{column_name}' is
{average:.2f}.")
        else:
            print(f"No numeric values found in column '{column_name}'.")

    except FileNotFoundError:
        print("File not found. Please check the file path and try again.")
    except Exception as e:
        print(f"An error occurred: {e}")

if __name__ == "__main__":
    calculate_column_average()
```

Output:

Create a spreadsheet and save it as a CSV file.



```
Python Console
Enter the path of the CSV file: >? C:\Users\Lenovo\Downloads\Untitled spreadsheet - Sheet1 (1).csv
Enter the column name to calculate the average: >? age
The average of the values in column 'age' is 26.33.

>>>
```

Practical 5:

QUES 5: Write a program in python that reads an Excel file and prints the data in a tabular format.

Aim: reading the entries in an excel file

Code:

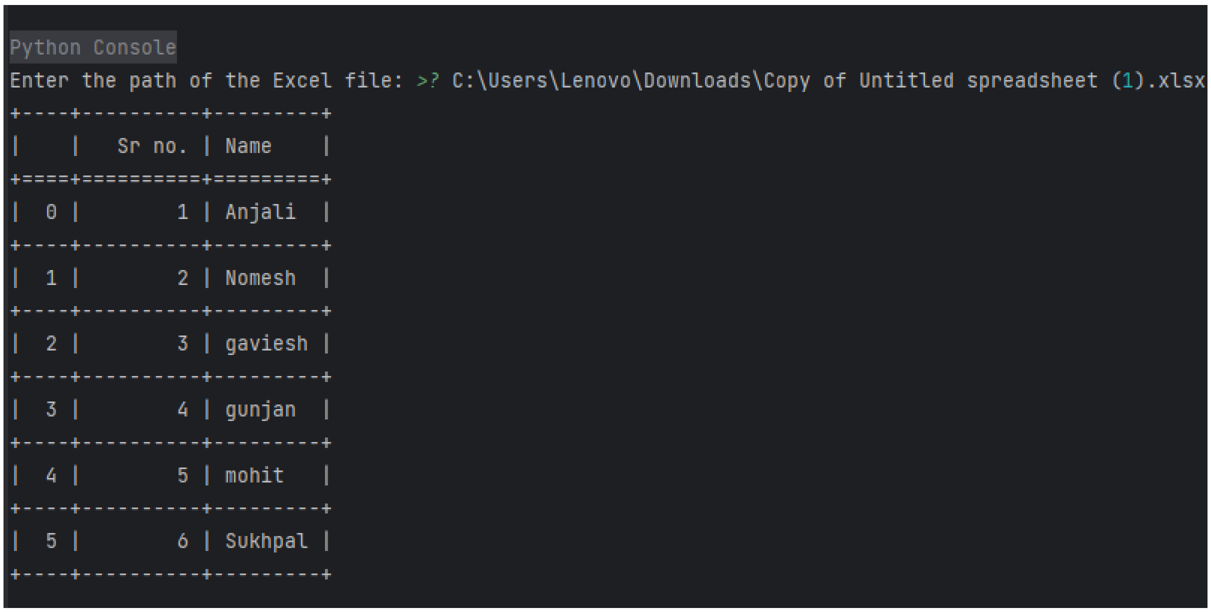
```
import pandas as pd
from tabulate import tabulate

def read_excel_file():
    file_path = input("Enter the path of the Excel file: ")

    try:
        # Read the Excel file
        df = pd.read_excel(file_path)

        # Print the data in a tabular format
        print(tabulate(df, headers='keys', tablefmt='grid'))
    except FileNotFoundError:
        print("File not found. Please check the file path and try again.")
    except Exception as e:
        print(f"An error occurred: {e}")

if __name__ == "__main__":
    read_excel_file()
```

Output:
First install packages:

Write in your console:

```
pip install pandas
pip install tabulate
pip install openpyxl
```

now run the file....



```
Python Console
Enter the path of the Excel file: >? C:\Users\Lenovo\Downloads\Copy of Untitled spreadsheet (1).xlsx
+----+----------+---------+
|    |   Sr no. | Name    |
+====+==========+=========+
|  0 |        1 | Anjali  |
+----+----------+---------+
|  1 |        2 | Nomesh  |
+----+----------+---------+
|  2 |        3 | gaviesh |
+----+----------+---------+
|  3 |        4 | gunjan  |
+----+----------+---------+
|  4 |        5 | mohit   |
+----+----------+---------+
|  5 |        6 | Sukhpal |
+----+----------+---------+
```