

Name : Jayesh Chaudhari

PYTHON ASSIGNMENT

Programming Task

Task 1: Calculate Area with Conditions

Write a Python function calculate_area that takes two parameters: length and width. It should calculate and return the area of a rectangle. However, add a condition: if the length is equal to the width, return "This is a square!" instead of the area. Then, write a program to input values for length and width from the user and call the calculate_area function to display either the area or the message.

```
In [1]: def calculate_area(length, width):
        if length == width:
            return "This is a square!"
        else:
            area = length * width
            return area

def main():
    try:
        length = float(input("Enter the length of the rectangle: "))
        width = float(input("Enter the width of the rectangle: "))

        result = calculate_area(length, width)

        print(result)
    except ValueError:
        print("Please enter valid numerical values for length and width.")

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

Enter the length of the rectangle: 20
Enter the width of the rectangle: 30
600.0

Task 2: Generate Fibonacci Series

Problem Statement:

Write a Python program that generates the Fibonacci sequence up to a specified number of terms, n. The Fibonacci sequence starts with 0 and 1, and each subsequent number in the sequence is the sum of the two preceding numbers (e.g., 0, 1, 1, 2, 3, 5, 8, ...). Prompt the user to enter the number of terms (n) they want in the sequence and then display the Fibonacci sequence up to that number of terms.

```
In [3]: def generate_fibonacci(n):
        fibonacci_sequence = [0, 1]

        while len(fibonacci_sequence) < n:
            next_number = fibonacci_sequence[-1] + fibonacci_sequence[-2]
            fibonacci_sequence.append(next_number)

        return fibonacci_sequence

def main():
    try:
        n = int(input("Enter the number of terms for the Fibonacci sequence: "))

        if n <= 0:
            print("Please enter a positive integer.")
        else:
            fibonacci_sequence = generate_fibonacci(n)
            print(f"Fibonacci sequence up to {n} terms: {fibonacci_sequence}")
    except ValueError:
        print("Please enter a valid integer for the number of terms.")

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

Enter the number of terms for the Fibonacci sequence: 10
Fibonacci sequence up to 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

Task 3: MySQL Database Operations with Python

Problem Statement:

Your task is to write a Python program that accomplishes the following: First create a database , table and add these column 'student_id', 'first_name', 'last_name', 'age', 'grade'. Connects to your MySQL database with python. Inserts a new student record into the "students" table with the following details: First Name: "Alice" Last Name: "Smith" Age: 18 Grade: 95.5 Updates the grade of the student with the first name "Alice" to 97.0. Deletes the student with the last name "Smith." Fetches and displays all student records from the "students" table.

```
In [ ]: pip install mysql-connector-python
```

```
In [7]: import mysql.connector
```

```
In [8]: # Function to create the "students" table
def create_table(cursor):
    create_table_query = """
    CREATE TABLE IF NOT EXISTS students (
        student_id INT AUTO_INCREMENT PRIMARY KEY,
        first_name VARCHAR(50),
        last_name VARCHAR(50),
        age INT,
        grade FLOAT
    )
    """
    cursor.execute(create_table_query)
```

```
In [9]: # Function to insert a new student record
def insert_student(cursor, first_name, last_name, age, grade):
    insert_query = """
    INSERT INTO students (first_name, last_name, age, grade)
    VALUES (%s, %s, %s, %s)
    """
    values = (first_name, last_name, age, grade)
    cursor.execute(insert_query, values)
```

```
In [10]: # Function to update the grade of a student
def update_grade(cursor, first_name, new_grade):
    update_query = """
    UPDATE students
    SET grade = %s
    WHERE first_name = %s
    """
    values = (new_grade, first_name)
    cursor.execute(update_query, values)
```

```
In [11]: # Function to delete a student by last name
def delete_student(cursor, last_name):
    delete_query = """
    DELETE FROM students
    WHERE last_name = %s
    """
    values = (last_name,)
    cursor.execute(delete_query, values)
```

```
In [12]: # Function to fetch and display all student records
def fetch_and_display_students(cursor):
    select_all_query = "SELECT * FROM students"
    cursor.execute(select_all_query)
    students = cursor.fetchall()

    if not students:
        print("No student records found.")
    else:
        print("Student records:")
        for student in students:
            print(student)

def main():
    # Connect to the MySQL database
    try:
        connection = mysql.connector.connect(
            host="your_host",
            user="your_user",
            password="your_password",
            database="your_database"
        )
        cursor = connection.cursor()

        # Create the "students" table if not exists
        create_table(cursor)

        # Insert a new student record
        insert_student(cursor, "Alice", "Smith", 18, 95.5)

        # Update the grade of the student with the first name "Alice"
        update_grade(cursor, "Alice", 97.0)

        # Delete the student with the last name "Smith"
        delete_student(cursor, "Smith")

        # Fetch and display all student records
        fetch_and_display_students(cursor)

        # Commit the changes and close the connection
        connection.commit()
        cursor.close()
        connection.close()
    except mysql.connector.Error as err:
        print(f"Error: {err}")

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

Student records: (1, 'Alice', 'Smith', 18, 97.0)

```
In [ ]:
```