

Name : Prabhuthvam Chakali

Email : prabhuthvamc@gmail.com

Contact No : 9391921717

Github Link :

<https://github.com/Prabhuthvam/samata/tree/main>



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.

Solution:

```
# user input
```

```
length = float(input("Enter the length: "))
```

```
width = float(input("Enter the width: "))
```

```
# Checking the condition
```

```
if length == width:
```

```
print("This is a square!")
```

else:

```
area = length * width
```

```
print("The area of the rectangle is:", area)
```

Example Output

```
'''
```

Enter the length: 5

Enter the width: 10

The area of the rectangle is: 50.0

Enter the length: 5

Enter the width: 5

This is a square!

```
'''
```

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.

Solution:

```
# user input
```

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

```
a, b = 0, 1
```

```
fibonacci_sequence = [a, b]
```

```
for _ in range(2, n):
```

```
    a, b = b, a + b
```

```
    fibonacci_sequence.append(b)
```

```
print("Fibonacci sequence up to", n, "terms:")
```

```
print(fibonacci_sequence)
```

```
# Example Output
```

```
'''
```

```
Enter the number of terms for the Fibonacci sequence: 5
```

```
Fibonacci sequence up to 5 terms:
```

```
[0, 1, 1, 2, 3]
```

```
Enter the number of terms for the Fibonacci sequence: 2
```

```
Fibonacci sequence up to 2 terms:
```

```
[0, 1]
```

```
'''
```

Task 3: MySQL Database Operations with Python (Compulsory) 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.

Solution:

```
import mysql.connector
```

```
mycon=mysql.connector.connect(host="localhost",user="root",passwd="root")
```

```
if mycon.is_connected():
```

```
    print("sucessfully connected")
```

```
cursor = mycon.cursor()
```

```
#creating database
```

```
cursor.execute("CREATE DATABASE school")
```

```
# using "school" database
```

```
cursor.execute("USE school")
```

```
#Creating table and adding columns
```

```
cursor.execute("CREATE TABLE students ('"student_id INT, '"first_name VARCHAR(255), '"  
"last_name VARCHAR(255), '"age INT, '"grade FLOAT)")
```

Task 2: Inserting a new student record

```
cursor.execute("INSERT INTO students (student_id, first_name, last_name, age, grade) VALUES  
(%s, %s, %s, %s,%s)",
```

```
    (11,"Alice", "Smith", 18, 95.5))
```

```
mycon.commit()
```

Task 3: Updating the grade of the student with the first name "Alice"

```
cursor.execute("UPDATE students SET grade = %s WHERE first_name = %s", (97.0, "Alice"))
```

```
mycon.commit()
```

#Task 4: Deleting the student with the last name "Smith"

```
cursor.execute("DELETE FROM students WHERE last_name = %s", ("Smith",))
```

```
mycon.commit()
```

Task 5: Fetching and display all student records

```
cursor.execute("SELECT * FROM students")
```

```
students = cursor.fetchall()
```

```
print("Student Records:")
```

```
for student in students:
```

```
    print(student)
```

Closing the cursor and database connection

```
cursor.close()
```

mycon.commit()

Students Table Data

The screenshot shows the MySQL Workbench interface. The 'Query 1' window displays the SQL query: `SELECT * FROM school.students;`. The 'Result Grid' shows the following data:

student_id	first_name	last_name	age	grade
11	Alice	Smith	18	97

The 'Output' window shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	09:02:09	SELECT * FROM school.students LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Student Table Data After Deletion

The screenshot shows the MySQL Workbench interface after deleting the 'students' table. The 'Query 1' window displays the SQL query: `SELECT * FROM school.students;`. The 'Result Grid' is empty. The 'Output' window shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	09:02:09	SELECT * FROM school.students LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
2	09:03:56	DROP DATABASE 'school'	1 row(s) affected	0.047 sec
3	09:07:23	SELECT * FROM school.students LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec