

Assignment 9.1

Task 1: Write a Python script to display "Hello, Python!" and also demonstrate the use of comments.

Prog:

```
print("Hello, Python!")
```

Use of Comments in Python:

Comments in python are used to annotate code to make it easier to understand for others or us in the future. They are ignored by the Python interpreter when running the code.

- Enhance code readability.
- Explaining code to other.
- Understanding code if studied after some time.
- Documenting the steps and needs for a function.
- Sharing code with fellow developers.
- Collaborating with multiple people.

Task 2: Create a Python program that uses different variable types (integer, float, string, boolean) and prints their types.

Prog:

```
price = 300
discount = 25.5
product='Tshirt' have_collar = True
print("Product name is", product)
print("Price of the Product is", price)
print("Discount for the product is", discount)
print("Have collar", have_collar)
print(type(price))
print(type(discount))
print(type(product))
print(type(have_col))
```

Output:

```
Product name is Tshirt Price of the Product is 300
Discount for the product is 25.5
Have collar True
Type of product <class 'int'>
<class 'float'>
```

<class 'str'>

<class 'bool'>

Assignment 9.2

Task 1: Create a Python list and demonstrate list slicing and appending new elements.

Prog:

```
#creating a list
nums = [1,2,3,4,5,6,7]
print(nums)

#slicing first four numbers
first_four = nums[:3]
print(first_four)

#slicing middle part of list
middle_part = nums[2:5]
print(middle_part)

#appending new elements into list
nums.append(8)
nums.append(9)
print("Updated list is", nums)
```

Output:

```
[1, 2, 3, 4, 5, 6, 7]
[1, 2, 3]
[3, 4, 5]
Updated list is [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Task 2: Write a Python script to create, update, and print a dictionary containing personal information (like name, age, city).

Prog:

```
#creating a dictionary
emp = {'name':'rahul', 'age':23, 'city':'hyderabad'}
print(emp)

#updating the dictionary
emp['empID']=25
print("updated dictionary is", emp)
```

Output:

```
{'name': 'rahul', 'age': 23, 'city': 'hyderabad'}
```

```
Updated dictionary is {'name': 'rahul', 'age': 23, 'city': 'hyderabad', 'empID': 25}
```

Assignment 9.3

Task 1: Write a function in Python that calculates and returns the factorial of a number provided as an argument.

Prog:

```
Def factorial(n):  
    if n<1:  
        return 1  
    else:  
        return n* factorial(n-1)  
num = 5  
print("Result is", factorial(num))
```

Output:

Result is 120

Task 2: Create a script using Python's range function to print all even numbers between 1 and 20.

Prog:

```
for i in range(2,20,2):  
    print(i)
```

Output:

2
4
6
8
10
12
14
16
18

Assignment 9.4

Task 1: Write a Python program to read a file and print its content line by line.

Prog:

```
def read_file(fname):
    try:
        with open(fname, 'r') as file:
            for line in file:
                print(line.rstrip())
    except FileNotFoundError:
        print("Error: file not found")

fname = 'new.txt'
read_file(fname)
```

Output:

```
Hello
How are you
```

Task 2: Create a Python class named Rectangle with attributes length and breadth and methods to calculate area and perimeter.

Prog:

```
Class Rectangle:
    def __init__(self, width, length):
        self.width = width
        self.length = length
    def cal_area(self):
        return (self.width * self.length)
    def cal_perimeter(self):
        return 2* (self.width + self.length)

rect1 = Rectangle(7,10)
print(f"Area of rectangle is {rect1.cal_area}")
print(f"Perimeter of rectangle is {rect1.cal_perimeter}")
```

Output:

```
Area of rectangle is 70 Perimeter of rectangle is 34
```

Assignment 9.5

Task 1: Use Python's re module to find all occurrences of the word "Python" in a given text.

Prog:

```
import re

def find_occurrences(text):
    occurrences = re.findall(r'\bpython\b', text, flags=re.IGNORECASE)
    return occurrences

example_text="""Python is widely used
                python was created by
                Python is easy to use"""

py_occurrences = find_occurrences(example_text)
print(len(py_occurrences))
```

Output:

3

Task 2: Create a list comprehension in Python to generate squares of even numbers between 1 to 10.

Prog:

```
list1 = [print(i*i) for i in range(2,10,2)]
```

Output:

4
16
36
64

Assignment 9.6

Task 1: Write a simple Python function to add two numbers and then write a pytest test case to test this function.

Prog1:

```
def add_nums(a,b):  
    return a + b
```

Prog2:

```
import pytest  
  
def test_add_nums():  
    result = add_nums(3,5)  
    assert result == 8  
  
    result = add_nums(-2, 7)  
    assert result == 5  
  
    result = add_nums(0, 10)  
    assert result == 10  
  
if __name__ == "__main__":  
    pytest.main()
```

Task 2: Write a pytest test case to check if an exception is raised for a function that divides two numbers.

Prog1:

```
def divide_nums(a, b):  
    if b == 0:  
        raise ValueError("cannot divide by zero")  
  
    return a / b
```

Prog2:

```
import pytest  
  
def test_divide_nums():  
    result = divide_nums(10,2)  
    assert result == 5.0  
  
    with pytest.raises (ValueError):  
        assert divide_nums(10,0)
```



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

Assignment 9.7

Task 1: Write SQL queries to create a table named 'Users' and insert multiple records into it.

Prog:

```
import sqlite3

conn = sqlite3.connect('sampledb.db')

cursor = conn.cursor()

create_table = """

CREATE TABLE users ( id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT NOT
NULL, email TEXT NOT NULL,

ceated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

cursor.execute(create_table)

conn.commit

users_data = [ ('rahul','rahul@email.com'), ('rohot', 'rohot@email.com'),
('dhoni','dhoni@email.com'))

insert_query = ""

INSERT INTO USERS (username, email) VALUES ('sachin','sachin@email.com')"""

cursor.executemany(insert_query, users_data)

conn.commit()
```

Task 2: Write a SQL query to update a record in the 'Users' table and then delete a record from it.

Prog:

```
update_query = """

UPDATE users

SET email = rahul@email.com

WHERE username = rahul"""

new_email = 'joshi@email.com'

user_id_update = 1

cursor.execute(update_query, (new_email,user_id_update))

conn.commit()

#delete a record form the user table

delete_query="""

DELETE FROM users
```

```
WHERE username = 'joshi'
```

```
"""
```

```
user_id_delete = 2
```

```
cursor.execute(delete_query, (user_id_delete,))
```

```
conn.commit()
```

Assignment 9.8

Task 1: Create a Python script to connect to a MySQL database and fetch records from a table.

Prog:

```
import mysql.connector

def fetch_records():
    try:
        conn = mysql.connector.connect(
            host='host_name', database='mydatabase', user='admin',
            password='*****')
        if conn.is_connected():
            print('connected to MySQL database')
        cursor = conn.cursor()
        query = "SELECT * FROM table1" cursor.execute(query)
        records = cursor.fetchall()
        for record in records:
            print(record)
        except mysql.connector.Error as e:
            print(f"Error connecting to MySQL database:{e}")
        finally:
            if connection.is_connected():
                cursor.close()
                connection.close()
        print("MySQL.connection is closed")
        host = 'localhost'
        database = 'mydatabase'
        user = 'admin'
        password = '*****'
        table_name = 'table1'
        connection = connect_to_mysql(host, database, user, password )
        if connection:
            fetch_record(connection, table_name)
```

else:

print("connection failed")

Task 2: Write a Python script to insert a new record into a database table using SQLite3.

Prog:

```
import sqlite3

conn = sqlite3.connect('mydatabase.db')

cursor = conn.cursor()

cursor.execute("CREATE TABLE IF NOT EXISTS users( id INTEGER PRIMARY KEY, name TEXT NOT NULL, age INTEGER)")

new_user = ('rahul', 23)

cursor.execute('INSERT INTO users (name, age) VALUES (?,?)', new_user)

conn.commit()

cursor.close()

conn.close()

print("New record inserted successfully")
```