20 Python Programs for Class 12 Computer Science Project by Sammy Siddique

1. Python Program for Stack Implementation

Program Code:

```
class Stack:
   def __init__(self):
      self.stack = []

   def push(self, item):
      self.stack.append(item)

   def pop(self):
      return self.stack.pop() if self.stack else "Stack Underflow"

   def display(self):
      print("Stack:", self.stack)

s = Stack()
s.push(5)
s.push(10)
print("Popped Element:", s.pop())
s.display()
```

Output:

```
Popped Element: 10

Stack:[5]
```

2. Python Program for MySQL Connection

Program Code:

```
import mysql.connector

connection = mysql.connector.connect(
   host="localhost", user="root", password="password", database="school"
)
cursor = connection.cursor()
cursor. execute("CREATE TABLE IF NOT EXISTS students (id INT, name VARCHAR(50))")
cursor.execute("INSERT INTO students (id, name) VALUES (1, 'Sameer Siddique')")
```

```
connection.commit()
cursor.execute("SELECT * FROM students")
for row in cursor.fetchall():
   print(row)
connection.close()
```

Output:

(1,'Sameer Siddique')

3. Python Program to Reverse a String

Program Code:

```
def reverse_string(s):
   return s[::-1]

string = " Sameer hackathon 2024"
print("Original String:", string)
print("Reversed String:", reverse_string(string))
```

Output:

Original String : Sameer hackathon 2024

Reversed String: 4204 nohtakcah reemaS

4. Python Program to Check Prime Number

Program Code:

```
def is_prime(n):
   if n < 2:
      return False
   for i in range(2, int(n ** 0.5) + 1):
      if n % i == 0:
         return False
   return True

number = 29
print(f"{number} is Prime:", is_prime(number))
```

Output:

29 is prime:True

5. Python Program for Bubble Sort

Program Code:

```
def bubble_sort(arr):
    for i in range(len(arr)):
        for j in range(len(arr) - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
    return arr

arr = [5, 1, 4, 2, 8]
print("Sorted Array:", bubble_sort(arr))
```

Output:

Sorted Array: [1,2,4,5,8]

6. Python Program for Fibonacci Sequence

Program Code:

```
def fibonacci(n):
    a, b = 0, 1
    for _ in range(n):
        print(a, end=" ")
        a, b = b, a + b

fibonacci(8)
```

Output:

0 1 1 2 3 5 8 1 3

7. Python Program for Factorial Calculation

Program Code:

```
def factorial(n):
    return 1 if n == 0 else n * factorial(n - 1)

number = 6
print(f"Factorial of {number}:", factorial(number))
```

Output:

Factorial of 6 : 720

8. Python Program to Check Palindrome

Program Code:

```
def is_palindrome(string):
    return string == string[::-1]

word = "radar"
print(f"{word} is Palindrome:", is_palindrome(word))
```

Output:

radar is palindrome: True

9. Python Program for File Handling

Program Code:

```
with open("example.txt", "w") as file:
    file.write("Relations & Functions")

with open("example.txt", "r") as file:
    content = file.read()
    print("File Content:", content)
```

Output:

File content : Relations & Functions

10. Python Program for Armstrong Number

Program Code:

```
def is_armstrong(n):
    num_str = str(n)
    return n == sum(int(digit) ** len(num_str) for digit in num_str)

number = 153
print(f"{number} is Armstrong:", is_armstrong(number))
```

Output:

153 is Armstrong : True

11. Python Program for Sum of Digits

Program Code:

```
def sum_of_digits(n):
    return sum(int(digit) for digit in str(n))

number = 12345
print(f"Sum of Digits of {number}:", sum_of_digits(number))
```

Output:

Sum of Digits of 12345 : 15

12. Python Program for GCD (Greatest Common Divisor)

Program Code:

```
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

print("GCD of 48 and 18:", gcd(48, 18))
```

Output:

GCD of 48 and 18 : 6

13. Python Program for Leap Year Check

Program Code:

```
def is_leap_year(year):
    return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)

year = 2024
print(f"{year} is Leap Year:", is_leap_year(year))
```

Output:

2024 is leap year

14. Python Program to Reverse a Stack

Program Code:

```
def reverse_stack(stack):
    if not stack:
```

```
    return []
  temp = stack.pop()
  reversed_stack = reverse_stack(stack)
  reversed_stack.append(temp)
  return reversed_stack

stack = [1, 2, 3]
print("Reversed Stack:", reverse_stack(stack))
```

Output:

Reversed stack: [3,2,1]

15. Python Program for Perfect Number

Program Code:

```
def is_perfect(n):
  return n == sum(i for i in range(1, n) if n % i == 0)

number = 28
print(f"{number} is Perfect Number:", is_perfect(number))
```

Output:

28 is perfect Number : True

16. Python Program to Count Vowels

Program Code:

```
def count_vowels(s):
  return sum(1 for char in s.lower() if char in "aeiou")

text = "Dav Public School Kathara."
print("Number of Vowels:", count_vowels(text))
```

Output:

Number of Vowels : 8

17. Python Program for Simple Calculator

Program Code:

```
def calculator(a, b, op):
  if op == '+': return a + b
  if op == '-': return a - b
```

```
    if op == '*': return a * b
    if op == '/': return a / b if b != 0 else "Division by Zero Error"

print("15 + 10 =", calculator(15, 10, '+'))
```

Output:

```
15 + 10 = 25
```

18. SQL Query to Create Database and Table 'Event'

```
-- Create the database

CREATE DATABASE EventManagement;


-- Use the newly created database

USE EventManagement;


-- Create the "Event" table with constraints

CREATE TABLE Event (

    eventid INT AUTO_INCREMENT PRIMARY KEY,  -- Event ID with auto increment

    event VARCHAR(255) NOT NULL,           -- Event name (NOT NULL constraint)

    location VARCHAR(255) DEFAULT 'Ranchi',  -- Location with default value

    clientid INT NOT NULL,                 -- Client ID (NOT NULL constraint)

    eventdate DATE NOT NULL,               -- Event Date (NOT NULL constraint)

    size INT,                              -- Size (Adjust according to requirement)

    FOREIGN KEY (clientid) REFERENCES client(clientid)  );     -- Foreign Key from the parent
                                                                  'client' table

DESCRIBE Event;
```

Output :

```
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| eventid   | int          | NO   | PRI | NULL    | auto_increment |
| event     | varchar(255) | NO   |     | NULL    |                |
| location  | varchar(255) | YES  |     | Ranchi  |                |
| clientid  | int          | NO   | MUL | NULL    |                |
| eventdate | date         | NO   |     | NULL    |                |
| size      | int          | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
```

19. SQL Query to insert values into the Event table and print the output :

```
-- Insert values into the Event table
INSERT INTO Event (event, location, clientid, eventdate, size)
VALUES ('Tech Conference', 'Ranchi', 101, '2025-02-10', 200),
 ('Farewell Function', 'DAV KATHARA', 102, '2025-03-15', 500),
('Art Exhibition', 'Dhanbad', 103, '2025-04-05', 150);
-- Select all records from the Event table
SELECT * FROM Event;
```

Output :

```
+---------+------------------+---------------+----------+------------+------+
| eventid | event            | location      | clientid | eventdate  | size |
+---------+------------------+---------------+----------+------------+------+
|       1 | Tech Conference  | Ranchi        |      101 | 2025-02-10 |  200 |
|       2 | Farewell Function| DAV KATHARA   |      102 | 2025-03-15 |  500 |
|       3 | Art Exhibition   | Dhanbad       |      103 | 2025-04-05 |  150 |
+---------+------------------+---------------+----------+------------+------+
```

20.Python Program to Count Words in a File

```
# Program to count the number of words in a file

def count_words_in_file(filename):
    try:
        with open(filename, "r") as file:
            content = file.read()
            words = content.split()
            print(f"File Content:\n{content}")
            print(f"\nTotal Number of Words: {len(words)}")
    except FileNotFoundError:
        print("The file does not exist.")


# Writing updated content to the file

filename = "sample.txt"

with open(filename, "w") as file:
    file.write("This is Sameer Siddique of class 12th/A of Science Department from Dav Kathara.\n")
    file.write("This is Computer Science project given by Jaipal sir.")


# Reading from the file and counting words

count_words_in_file(filename)
```

Output :

```
File Content:
This is Sameer Siddique of class 12th/A of Science Department from Dav Kathara.
This is Computer Science project given by Jaipal sir.

Total Number of Words: 19
```

Github profile: