

CSE PRACTICAL **QUESTIONS**

Name – Nipul Rajput

Roll No. – 2024UCA1883

Branch – CSAI- 2

Group - 1

Q.1 Write a Python program to print "Hello, World!"; Write a Python program to calculate the area of a circle given the radius.

```
Hello World
Enter radius : 5
Area of circle is 78.5
```

```
print('Hello World')
n=int(input("Enter radius : "))
a=3.14*n*n
print("Area of circle is ",a)
```

Q.2 Write a Python program to check if a number is even or odd.; Implement a simple calculator using conditional statements; Write a Python program to print the Fibonacci series using a for loop.

To find if even or odd:

```
Enter a number: 6
6 is even.
```

```
def check_even_odd(number):
    if number % 2 == 0:
        print(number, " is even.")
    else:
        print(number, " is odd.")
num = int(input("Enter a number: "))
check_even_odd(num)
```

To implement a calculator :

```

while True:
    choice = input("Enter choice (1/2/3/4): ")
    if choice in ('1', '2', '3', '4'):
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))
        if choice == '1':
            result = num1 + num2
            print(f"The result is: {result}")
        elif choice == '2':
            result = num1 - num2
            print(f"The result is: {result}")
        elif choice == '3':
            result = num1 * num2
            print(f"The result is: {result}")
        elif choice == '4':
            if num2 == 0:
                print("Error! Division by zero.")
            else:
                result = num1 / num2
                print(f"The result is: {result}")
        else:
            print("Invalid Input")
        next_calculation = input("Do you want to perform another calculation? (yes/no): ")
        if next_calculation.lower() != 'yes':
            break
    print("Thank you for using the calculator!")

```

```

1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4): 1
Enter first number: 1
Enter second number: 2
The result is: 3.0
Do you want to perform another calculation? (yes/no): yes
Enter choice (1/2/3/4): 2
Enter first number: 2
Enter second number: 1
The result is: 1.0
Do you want to perform another calculation? (yes/no): yes
Enter choice (1/2/3/4): 3
Enter first number: 2
Enter second number: 3
The result is: 6.0
Do you want to perform another calculation? (yes/no): yes
Enter choice (1/2/3/4): 4
Enter first number: 6
Enter second number: 2
The result is: 3.0
Do you want to perform another calculation? (yes/no): |

```

Fibonacci series using for loop:

```
n= int(input("Enter the number of terms: "))
a, b = 0, 1
print("Fibonacci series:")
for i in range(n):
    print(a, end=" ")
    a, b = b, a + b
```

```
Enter the number of terms: 4
Fibonacci series:
0 1 1 2
```

Q.3 Implement a function to check if a given string is a palindrome.; Perform various operations on lists (e.g., sorting, slicing).; Use dictionaries to store and retrieve student grades.

Function to Check if a Given String is a Palindrome

```
def is_palindrome(s):
    s = s.replace(" ", "").lower()
    return s == s[::-1]
string = input("Enter a string: ")
if is_palindrome(string):
    print(f"'{string}' is a palindrome.")
else:
    print(f"'{string}' is not a palindrome.")
```

```
Enter a string: nitin
'nitin' is a palindrome.
```

Various Operations on Lists

```

numbers = [1,2,3,4,5,6,7]
sorted_numbers = sorted(numbers)
reverse_sorted_numbers = sorted(numbers, reverse=True)
first_three = numbers[:3]
last_three = numbers[-3:]
middle_elements = numbers[1:5]
print("Original list:", numbers)
print("Sorted list (ascending):", sorted_numbers)
print("Sorted list (descending):", reverse_sorted_numbers)
print("First three elements:", first_three)
print("Last three elements:", last_three)
print("Middle elements (index 1 to 4):", middle_elements)

```

```

Original list: [1, 2, 3, 4, 5, 6, 7]
Sorted list (ascending): [1, 2, 3, 4, 5, 6, 7]
Sorted list (descending): [7, 6, 5, 4, 3, 2, 1]
First three elements: [1, 2, 3]
Last three elements: [5, 6, 7]
Middle elements (index 1 to 4): [2, 3, 4, 5]

```

Using Dictionaries to Store and Retrieve Student Grades

```

grades = {"Nipul": 85, "Roshan": 78, "Prankush": 92, "Bhavishya": 88, "Snehil": 76}
student_name = input("Enter the student's name to get their grade: ")
if student_name in grades:
    print(f"{student_name}'s grade is {grades[student_name]}")
else:
    print(f"No grade found for {student_name}")

```

```

Enter the student's name to get their grade: Bhavishya
Bhavishya's grade is 88

```

Q.4 Create a class to represent a book with attributes and methods.; Implement inheritance by creating subclasses for different types of books.; Write a generator function to generate the Fibonacci series.

Create a Class to Represent a Book with Attributes and Methods

```

class Book:
    def __init__(self, title, author, year):
        self.title = title
        self.author = author
        self.year = year

    def display_info(self):
        print(f"Title: {self.title}")
        print(f"Author: {self.author}")
        print(f"Year: {self.year}")

book1 = Book("1984", "George Orwell", 1949)
book1.display_info()

```

```

Title: 1984
Author: George Orwell
Year: 1949

```

Implement Inheritance by Creating Subclasses for Different Types of Books :

```

class AudioBook(Book):
    def __init__(self, title, author, year, duration):
        super().__init__(title, author, year)
        self.duration = duration

    def display_info(self):
        super().display_info()
        print(f"Duration: {self.duration} hours")

ebook = EBook("Digital Fortress", "Dan Brown", 1998, 5)
audiobook = AudioBook("The Hobbit", "J.R.R. Tolkien", 1937, 11.5)
print("EBook Information:")
ebook.display_info()
print("\nAudioBook Information:")
audiobook.display_info()

```

```

EBook Information:
Title: Digital Fortress
Author: Dan Brown
Year: 1998
File Size: 5 MB

```

```

AudioBook Information:
Title: The Hobbit
Author: J.R.R. Tolkien
Year: 1937
Duration: 11.5 hours

```

Generator Function to Generate the Fibonacci Series:

```

n= int(input("Enter the number of terms: "))
a, b = 0, 1
print("Fibonacci series:")
for i in range(n):
    print(a, end=" ")
    a, b = b, a + b

```

```
Enter the number of terms: 4
Fibonacci series:
0 1 1 2
```

Q.5 Use lambda functions, map, and filter to perform operations on a list.; Create a module that contains functions for mathematical operations.; Import and use functions from external packages (e.g., math, random).

Use Lambda Functions, map, and filter to Perform Operations on a List:

```
numbers = [1, 2, 3, 4, 5, 6, 7]
squared_numbers = list(map(lambda x: x ** 2, numbers))
print("Squared Numbers:", squared_numbers)
even_numbers = list(filter(lambda x: x % 2 == 0, numbers))
print("Even Numbers:", even_numbers)
doubled_numbers = list(map(lambda x: x * 2, numbers))
print("Doubled Numbers:", doubled_numbers)
```

```
Squared Numbers: [1, 4, 9, 16, 25, 36, 49]
Even Numbers: [2, 4, 6]
Doubled Numbers: [2, 4, 6, 8, 10, 12, 14]
```

Create a Module that Contains Functions for Mathematical Operations:

```
import math as mo
x = 10
y = 5
print(f"Addition of {x} and {y}: {mo.add(x, y)}")
print(f"Subtraction of {x} and {y}: {mo.subtract(x, y)}")
print(f"Multiplication of {x} and {y}: {mo.multiply(x, y)}")
print(f"Division of {x} and {y}: {mo.divide(x, y)}")
```

```
Addition of 10 and 5: 15
Subtraction of 10 and 5: 5
Multiplication of 10 and 5: 50
Division of 10 and 5: 2.0
```

Import and Use Functions from External Packages (math and random):

```

import math
import random
number = 16
square_root = math.sqrt(number)
print(f"Square root of {number} is {square_root}")
angle = math.radians(30)
sine_value = math.sin(angle)
print(f"Sine of 30 degrees is {sine_value}")
random_number = random.randint(1, 100)
print(f"Random number between 1 and 100: {random_number}")
random_float = random.random()
print(f"Random float between 0 and 1: {random_float}")

```

Square root of 16 is 4.0
 Sine of 30 degrees is 0.49999999999999994
 Random number between 1 and 100: 9
 Random float between 0 and 1: 0.7651081753846921

Q.6 Create and manipulate NumPy arrays.; Perform basic operations and indexing on arrays

<pre> import numpy as np array1 = np.array([1, 2, 3, 4, 5]) print("Array 1:", array1) array2 = np.arange(0, 10, 2) print("Array 2:", array2) zeros_array = np.zeros((3, 3)) print("Zeros Array:\n", zeros_array) ones_array = np.ones((2, 4)) print("Ones Array:\n", ones_array) random_array = np.random.rand(3, 3) print("Random Array:\n", random_array) array3 = np.array([10, 20, 30, 40, 50]) add_result = array1 + array3 print("Addition Result:", add_result) sub_result = array3 - array1 print("Subtraction Result:", sub_result) mul_result = array1 * 2 print("Multiplication Result:", mul_result) div_result = array3 / array1 print("Division Result:", div_result) elementwise_mul = array1 * array3 print("Element-wise Multiplication:", elementwise_mul) </pre>	<pre> Array 1: [1 2 3 4 5] Array 2: [0 2 4 6 8] Zeros Array: [[0. 0. 0.] [0. 0. 0.] [0. 0. 0.]] Ones Array: [[1. 1. 1. 1.] [1. 1. 1. 1.]] Random Array: [[0.25061373 0.57745503 0.52530922] [0.11742856 0.91314509 0.86396029] [0.08862629 0.85423109 0.67259522]] Addition Result: [11 22 33 44 55] Subtraction Result: [9 18 27 36 45] Multiplication Result: [2 4 6 8 10] Division Result: [10. 10. 10. 10. 10.] Element-wise Multiplication: [10 40 90 160 250] </pre>
---	---

Q.7 Implement string operations (e.g., concatenation, slicing).; Use regular expressions to validate email addresses.


```
# Concatenation
words = ["Python", "is", "fun"]
sentence = " ".join(words)
print("Joined String:", sentence)
#Slicing
text = "Hello, World!"
hello = text[:5]
print("Sliced Text (Hello):", hello)
world = text[7:12]
print("Sliced Text (World):", world)
alternate_chars = text[::2]
print("Alternate Characters:", alternate_chars)
reversed_text = text[::-1]
print("Reversed Text:", reversed_text)
```

Joined String: Python is fun
Sliced Text (Hello): Hello
Sliced Text (World): World
Alternate Characters: Hlo ol!
Reversed Text: !dlroW ,olleH

Q.8 Read data from a text file and perform operations.; Handle exceptions for file operations and input validation.

```
with open('data.txt', 'w') as file:
    file.write("Hello World\n")
    file.write("Python is great\n")
    file.write("Data Science is fascinating\n")
    file.write("File handling in Python\n")
def read_file(file_name):
    try:
        with open(file_name, 'r') as file:
            content = file.readlines() # Read all lines from the file
            return content
    except FileNotFoundError:
        print(f"Error: The file '{file_name}' does not exist.")
        return []
    except IOError:
        print("Error: An I/O error occurred.")
        return []
def perform_operations(lines):
    if not lines:
        print("No lines to process.")
        return
    line_count = len(lines)
    word_count = sum(len(line.split()) for line in lines)
    print(f"Total number of lines: {line_count}")
    print(f"Total number of words: {word_count}")
file_name = input("Enter the name of the file to read (e.g., data.txt): ")
while not file_name:
    file_name = input("Invalid input. Please enter the file name: ")
lines = read_file(file_name)
perform_operations(lines)
```

```
Enter the name of the file to read (e.g., data.txt): data.txt
Total number of lines: 4
Total number of words: 13
```