Pimpri Chinchwad Education Trust's

Pimpri Chinchwad College of Engineering

An Autonomous Institute

(Affiliated to Savitribai Phule Pune University)

Practical Assignment Submission

Department: MCA **Academic Year: 2023-24** **Semester: II**
Year: FYMCA **Course: Python Programming Lab**

PRN: 123M1H010 **Submission Date:** 22 /01/2024

Name of Student: Harshal Sanjay Bhamare
Practical Assignment No.: 01

---

1) Write a Python code to calculate the factorial of a given number.

Python Program:

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


num = int(input("Enter a number: "))
if num < 0:
   print("Factorial is not defined ")
else:
   result = factorial(num)
   print(f"The Factorial of {num} is: {result}")
```

Output:

```
Enter a number: 5
The Factorial of 5 is: 120
```

2) Create a Python program to check if a number is prime or not.

Python Program:

```
def Prime_check(number):
   is_prime = True
   for i in range(2,number):
      if number % i == 0:
            is_prime = False
    if is_prime:
        print("It is prime number")
    else:
        print("It is not a Prime Number")

n = int(input())
Prime_check(number=n)
```

Output:

```
6
It is not a Prime Number
```

3. Write a Python code to reverse a string.

Python Program:

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

original_string = "Hello World"
reversed_string = reverse_string(original_string)

print("Original String :",original_string)
print("Reversed String :",reversed_string)
```

Output:

```
Original String : Hello World
Reversed String : dlroW olleH
```

4. Use Python to create a simple calculator that can perform basic arithmetic operations.

Python Program:

```
while True:
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))
    operation = input("Enter the operation (+,-,*,/) ")
    result = None
    if operation == '+':
        result = num1 + num2
    elif operation == '-':
        result = num1 - num2
    elif operation == '*':
        result = num1 * num2
    elif operation == '/':
        if num2 != 0:
            result = num1 / num2
        else:
            print("Invalid")

    else:
        print("Invalid operations.")

    if result is not None:
        print(f"The result is {result}")
        break
```

Output:

```
Enter first number: 5
Enter second number: 6
Enter the operation (+,-,*,/) +
The result is 11.0
```

5. Implement a Python function to find the largest element in a list.

Python Program:

```
numbers = [10,5,6,8,4,9,16,25,4]
largest_element = max(numbers)
print(f"The largest element in the list is : {largest_eleme
```

Output:

```
The largest element in the list is : 25
```

6. Design a Python program to simulate a simple game of rock-paper-scissors.

Python Program:

```
import random

def get_user_choice():
    user_choice = input("Enter your choice (rock/paper/scissors): ").lower()
    while user_choice not in ['rock', 'paper', 'scissors']:
        print("Invalid choice. Please enter rock, paper, or scissors.")
        user_choice = input("Enter your choice (rock/paper/scissors): ").lower()
    return user_choice

def get_computer_choice():
    return random.choice(['rock', 'paper', 'scissors'])

def determine_winner(user_choice, computer_choice):
    if user_choice == computer_choice:
        return "It's a tie!"
    elif (
        (user_choice == 'rock' and computer_choice == 'scissors') or
        (user_choice == 'paper' and computer_choice == 'rock') or
        (user_choice == 'scissors' and computer_choice == 'paper')
    ):
        return "You win!"
    else:
        return "Computer wins!"

# Game:
while True:
    user_choice = get_user_choice()
    computer_choice = get_computer_choice()

    print(f"\nYour choice: {user_choice}")
    print(f"Computer's choice: {computer_choice}")

    result = determine_winner(user_choice, computer_choice)
    print(result)

    play_again = input("Do you want to play again? (yes/no): ").lower()
    if play_again == 'no':
        break

print("Thanks for playing!")
```

Output:

```
Enter your choice (rock/paper/scissors): rock

Your choice: rock
Computer's choice: scissors
You win!
Do you want to play again? (yes/no): no
Thanks for playing!
```

7. Develop a Python code to generate a random password with specified criteria.

Python Program:

```
import random
import string

def generate_pass(length=15):
    characters = string.ascii_letters + string.digits + string.punctuation
    password = ''.join(random.choice(characters) for _ in range(length))
    return password

password_length = int(input("Enter the desired password length."))
random_password = generate_pass(password_length)
```

```
print(f"Generated Password : {random_password}")
```

Output:

```
Enter the desired password length.8
Generated Password : q'ED"}&m
```

8. Create a Python script that calculates the area and perimeter of different geometric shapes based on user-provided dimensions.

Python Program:

```
import math

def rectangle_properties(length, width):
    area = length * width
    perimeter = 2 * (length + width)
    return area, perimeter
def square_properties(side_length):
    area = side_length ** 2
    perimeter = 4 * side_length
    return area, perimeter
def triangle_area(base, height):
    area = 0.5 * base * height
    return area

def triangle_perimeter(s1, s2, s3):
    perimeter = s1 + s2 + s3
    return perimeter

def circle_properties(radius):
    area = math.pi * (radius ** 2)
    perimeter = 2 * math.pi * radius
    return area, perimeter

print("Geometric Shape Calculator")
print("1. Rectangle")
print("2. Square")
print("3. Triangle")
print("4. Circle")

choice = input("Enter the number corresponding to the geometric shape (1/2/3/4): ")

if choice == '1':
    length = float(input("Enter the length of the rectangle: "))
    width = float(input("Enter the width of the rectangle: "))
    area, perimeter = rectangle_properties(length, width)
    shape_name = "Rectangle"
elif choice == '2':
    side_length = float(input("Enter the side length of the square: "))
    area, perimeter = square_properties(side_length)
    shape_name = "Square"
elif choice == '3':
    height = float(input("Enter the side height: "))
    base = float(input("Enter the base: "))
    area = triangle_area(base, height)
    s1 = float(input("Enter the side 1: "))
    s2 = float(input("Enter the side 2: "))
    s3 = float(input("Enter the side 3: "))
    perimeter = triangle_perimeter(s1, s2, s3)
    shape_name = "Triangle"
elif choice == '4':
    radius = float(input("Enter the radius of the circle: "))
    area, perimeter = circle_properties(radius)
    shape_name = "Circle"
else:
    print("Invalid input. Please enter a valid choice (1/2/3/4).")
    exit()

print(f"\n{shape_name} Properties:")
print(f"Area: {area}")
print(f"Perimeter: {perimeter}")
```

Output:

```
Geometric Shape Calculator
1. Rectangle
2. Square
3. Triangle
4. Circle
Enter the number corresponding to the geometric shape (1/2/3/4): 2
Enter the side length of the square: 5

Square Properties:
Area: 25.0
Perimeter: 20.0
```

9. Write a Python program that reads two numbers from the user, calculates their sum and difference, and prints the results with appropriate labels.

Python Program:

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
sum_result = num1 + num2
difference_result = num1 - num2
print(f"Sum of {num1} and {num2}: {sum_result}")
print(f"Difference of {num1} and {num2}: {difference_result}")
```

Output:

```
Enter the first number: 5
Enter the second number: 8
Sum of 5.0 and 8.0: 13.0
Difference of 5.0 and 8.0: -3.0
```

10. Create a Python script that converts a Fahrenheit temperature to Celsius and vice versa, using appropriate data types and conversion formulas.

Python Program:

```
def to_fahrenheit(cel):
   fahr = (cel*1.8) + 32
   return fahr

def to_celsius(fahr):
   cel = (fahr*1.8) - 32
   return cel

print("37 celsius in fahrenheit is: ", to_fahrenheit(37))
print("118 fahrenheit in celsius is: ", to_celsius(
```

Output:

```
37 celsius in fahrenheit is:  98.60000000000001
118 fahrenheit in celsius is:  180.4
```

11. Implement a Python function to sort a list of numbers using the bubble sort algorithm.

Python Program:

```
def bubble_sort(list1):
   length = len(list1)
   for i in range(length):
      for j in range(0, length-i-1):
         if list1[j]>list1[j+1]:
            temp = list1[j]
            list1[j] = list1[j+1]
            list1[j+1] = temp
   return list1

list1 = [1, 3, 6, 4, 2, 9, 7]

print("Given List: ", list1)
print("Sorted List: ", bubble_sort(list1))
```

Output:

```
Given List:  [1, 3, 6, 4, 2, 9, 7]
Sorted List:  [1, 2, 3, 4, 6, 7, 9]
```

12. Write a Python program that takes a list of numbers as input and prints the sum of all the even numbers in the list.

Python Program:

```
def sum_even(list1):
   sum = 0
   length = len(list1)
   for i in range (length):
      if (list1[i] % 2 == 0):
         sum = sum + list1[i]
   return sum

list1 = [1, 3, 6, 4, 2, 9, 7]

print("Sum of even numbers in the given list: " ,sum_even(list1))
```

Output:

```
Sum of even numbers in the given list:  12
```

13. Write a Python program that takes a list of strings and prints the count of each unique word in the list.

Python Program:

```
from collections import Counter

def count_unique_words(string_list):
   words = ' '.join(string_list).split()
   word_count = Counter(words)
   for word, count in word_count.items():
      print(f"{word}: {count}")

string_list = ["apple", "banana", "apple", "orange", "banana", "grape", "apple"]
count_unique_words(string_list)
```

Output:

```
apple: 3
banana: 2
orange: 1
grape: 1
```

14. Create a program that reverses a given list of strings and prints the reversed list.

Python Program:

```
list1 = ["Hello", "World", "Nice", "Day", "World"]
length = len(list1)
list_rev = []

for i in range (-1, -length-1, -1):
   list_rev.append(list1[i])

print("Original List: ", list1)
print("Reversed List: ", list_rev)
```

Output:

```
Original List:  ['Hello', 'World', 'Nice', 'Day', 'World']
Reversed List:  ['World', 'Day', 'Nice', 'World', 'Hello']
```

15. Write a Python program that defines two tuples of numbers and calculates the element-wise sum of the tuples.

Python Program:

```
def elementwise_sum(tuple1, tuple2):
   if len(tuple1) != len(tuple2):
      raise ValueError("Tuples must have the same length for element-wise sum.")

   result_tuple = tuple(a + b for a, b in zip(tuple1, tuple2))
   return result_tuple

tuple1 = (1, 2, 3, 4)
tuple2 = (5, 6, 7, 8)

result_sum = elementwise_sum(tuple1, tuple2)

print(f"Tuple 1: {tuple1}")
print(f"Tuple 2: {tuple2}")
print(f"Element-wise sum: {result_sum}")
```

Output:

```
Tuple 1: (1, 2, 3, 4)
Tuple 2: (5, 6, 7, 8)
Element-wise sum: (6, 8, 10, 12)
```

16. Create a program that checks if a given element exists in a tuple and prints whether it is present or not.

Python Program:

```
def check_element_in_tuple(element, input_tuple):
    if element in input_tuple:
        print(f"The element '{element}' is present in the tuple.")
    else:
        print(f"The element '{element}' is not present in the tuple.")

example_tuple = (1, 3, 'apple', 7, 'banana', 5.4)
element_to_check = 'banana'
check_element_in_tuple(element_to_check, example_tuple)
```

Output:

```
The element 'banana' is present in the tuple.
```

17. Write a Python program that takes a dictionary of student names and their corresponding scores and prints the student with the highest score.

Python Program:

```
def find_highest_scoring_student(scores_dict):
    if not scores_dict:
        print("Empty dictionary. No students to evaluate.")
        return

    highest_score = max(scores_dict.values())
    highest_scoring_students = [student for student, score in scores_dict.items() if score
== highest_score]

    if len(highest_scoring_students) == 1:
        print(f"The student with the highest score is: {highest_scoring_students[0]} with a
score of {highest_score}")
    else:
        print(f"There are multiple students with the highest score ({highest_score}): {',
'.join(highest_scoring_students)}")

student_scores = {'Alice': 85, 'Bob': 92, 'Charlie': 88, 'David': 92, 'Eva': 90}

find_highest_scoring_student(student_scores)
```

Output:

```
There are multiple students with the highest score (92): Bob, David
```

18. Create a program that merges two dictionaries and prints the resulting dictionary. If there are common keys, sum the values.

Python Program:

```
def merge_and_sum(dict1, dict2):
    result_dict = {}
    for key in set(dict1.keys()) | set(dict2.keys()):
        result_dict[key] = dict1.get(key, 0) + dict2.get(key, 0)

    return result_dict

dict1 = {'a': 10, 'b': 20, 'c': 30}
dict2 = {'b': 5, 'c': 15, 'd': 25}

result_dictionary = merge_and_sum(dict1, dict2)
print("Merged and summed dictionary:", result_dictionary)
```

Output:

```
Merged and summed dictionary: {'d': 25, 'a': 10, 'c': 45, 'b': 25}
```

19. Write a Python program that takes two sets of numbers and prints the union of the sets.

Python Program:

```
setA = {2, 4, 42, 54, 12, 43, 32}
setB = {42, 44, 12, 54, 12, 64, 23}

UnionSet = setA.union(setB)

print("Union of set A & B:", UnionSet)
```

Output:

```
Union of set A & B: {32, 64, 2, 4, 42, 43, 12, 44, 54, 23}
```

20. Create a program that checks if one set is a subset of another and prints the result.

Python Program:

```
Vehicles = {"Harley-Davidson", "Ferrari", "Pagani", "Ducati", "Royal-Enfield", "Porche"}
Cars = {"Ferrari", "Porche", "Pagani"}

print("Is Cars a subset of Vehicles? : ", Cars.issubset(Vehicles))
```

Output:

```
Is Cars a subset of Vehicles? :  True
```