SET 1

Practical 1

Ques: Write a program in python that converts temperatures from Fahrenheit to Celsius and vice versa.

Aim: to check temperature from fahrenheit to celsius using formula "**F = (C * 9/5) + 32**" .

Code:
```
fahrenheit = float(input("Enter temperature in fahrenheit: "))

celsius = (fahrenheit - 32)/1.8

print(str(fahrenheit )+ " degree Fahrenheit is equal\
to " + str(celsius ) + " degree Celsius." )
```

OUTPUT

Enter temperature in fahrenheit: 99

99.0 degree Fahrenheit is equalto 37.22222222222222 degree Celsius.

Practical 2:

Ques: Write a program in python that calculates the area and perimeter of a rectangle.

Aim: finding area and perimeter of a rectangle

Code:

```
a= int(input("enter value of a:"))
b= int(input("enter value of b:"))

Area= (a*b)
Perimeter = 2*(a+b)

print("Area of rectangle is" ,Area)
print("Perimeter of rectangle is" ,Perimeter)
```

OUTPUT

enter value of a:3

enter value of b:4

Area of rectangle is 12

Perimeter of rectangle is 14

Practical 3:

Ques: Write a python program that generates a random password of a specified length.

Aim: to generate a secure and strong random password

Code:

```
import random
import string

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

# Execution starts here
length = input("Enter the desired length of the password: ")

if length.isdigit():
    length = int(length)
    if length < 4:
```

```
        print("Password length should be at least 4.")
    else:
        password = generate_password(length)
        print("Generated password:", password)
else:
    print("Please enter a valid number for password length.")
```

OUTPUT

Enter the desired length of the password: 12

Generated password: p(&wNK^@.H+X

Practical 4:

Ques: Write a python program that calculates the average of a list of numbers.

Aim: to calculate average of numbers

Code:

```
def calculate_average(numbers):
    if len(numbers) == 0:  # Handle case where the list is empty
        return 0
    return sum(numbers) / len(numbers)


numbers_input = input("Enter a list of numbers separated by spaces: ")


numbers = [float(num) for num in numbers_input.split()]


average = calculate_average(numbers)


print(f"The average is: {average}")
```

OUTPUT

Enter a list of numbers separated by spaces: 3 5 2 1 8

The average is: 3.8

Practical 5:

Ques. Write a python program that checks if a given year is a leap year.

Aim: to check if an year contains 365 days or 366 days i.e. leap year or not

Code:

```python
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False


year = int(input("Enter a year: "))


if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

OUTPUT

Enter a year: 2024

2024 is a leap year.

Practical 6:

Ques. Write a python program that calculates the factorial of a number.

Aim: to calculate factorial of given number

Code:

```python
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result


num = int(input("Enter a number: "))
fact = factorial(num)


print(f"The factorial of {num} is: {fact}")
```

OUTPUT

Enter a number: 7

The factorial of 7 is: 5040

Practical 7:

Ques. Write a python program that checks if a given string is a palindrome.

Aim: to check if string is palindrome or not i.e ReversedString = OriginalString

Code:

```python
def is_palindrome(s):
    # Remove spaces and convert to lowercase for a case-insensitive comparison
    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.")
```

OUTPUT

Enter a string: racecar

'racecar' is a palindrome.

Practical 8:

Ques. A program that sorts a list of numbers in ascending or descending order.

Aim: to sort numbers in ascending or descending order

Code:

```
def sort_numbers(numbers, order='ascending'):
    if order == 'ascending':
        numbers.sort()  # Sort the list in ascending order
    elif order == 'descending':
        numbers.sort(reverse=True)  # Sort the list in descending order
    else:
        print("Invalid order specified. Choose 'ascending' or 'descending'.")
        return None
    return numbers


numbers_input = input("Enter a list of numbers separated by spaces: ")

numbers = [float(num) for num in numbers_input.split()]

order = input("Enter sorting order (ascending/descending): ").lower()
```

sorted_numbers = sort_numbers(numbers, order)

if sorted_numbers is not None:
    print(f"The sorted list ({order}): {sorted_numbers}")

OUTPUT

Enter a list of numbers separated by spaces: 3 4 2 1 6 5

Enter sorting order (ascending/descending): descending

The sorted list (descending): [6.0, 5.0, 4.0, 3.0, 2.0, 1.0]

Practical 9:

Ques. Write a python program that generates a multiplication table for a given number.

Aim: to get table of a given integer

Code:

```
def generate_multiplication_table(number):
    for i in range(1, 11):
        print(f"{number} x {i} = {number * i}")

number = int(input("Enter a number to generate its multiplication table: "))

generate_multiplication_table(number)
```

OUTPUT

Enter a number to generate its multiplication table: 5

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

Practical 10:

Ques. Write a python program that converts a given number from one base to another.

Aim: converting number from decimal base to another base

Code:

```
# Function to convert decimal to binary
def decimal_to_binary(dec):
        decimal = int(dec)


        print(decimal, " in Binary : ", bin(decimal))


# Function to convert decimal to octal
def decimal_to_octal(dec):
        decimal = int(dec)


        print(decimal, "in Octal : ", oct(decimal))


# Function to convert decimal to hexadecimal
def decimal_to_hexadecimal(dec):
        decimal = int(dec)
```

```
        print(decimal, " in Hexadecimal : ", hex(decimal))

dec = float(input("Enter number:"))
decimal_to_binary(dec)
decimal_to_octal(dec)
decimal_to_hexadecimal(dec)
```

OUTPUT

```
enter number:55
55  in Binary :  0b110111
55 in Octal :  0o67
55  in Hexadecimal :  0x37
```