**ASSESSMENT -1**

1. Write a Python program to calculate the area of a rectangle given its length and width.

def areaofrectangle (len, wid):

area = len \* wid

return area

def main():

len = float(input(“Enter length value:”))

wid = float(input(“Enter width value:”))

area = areaofrectangle(len,wid)

print(“area of a rectangle:” +area)

if \_\_ name \_\_ == “\_\_main\_\_” :

main()

2.Write a program to convert miles to kilometres.

def milestokilometers(m):

Kilometers = miles \* 1.60934

Return kilometres

def main()

If \_\_ name \_\_ == “\_\_main\_\_”:

Miles=float(input(“enter miles:”))

Kilometres=milestokilometers(Miles)

Print(kilometers)

3. Write a function to check if a given string is a palindrome.

def palindrome(s):

return s==s[::-1]

def main()

if \_\_ name\_\_ == “\_\_main\_\_”:

string = input(“Enter string:”)

if palindrome(string):

print(“palindrome”)

else:

print(“not palindrome”)

4. Write a Python program to find the second largest element in a list.

def findLargest(arr):

secondLargest = 0

largest = min(arr)

for i in range(len(arr)):

if arr[i] > largest:

secondLargest = largest

largest = arr[i]

else:

secondLargest = max(secondLargest, arr[i])

return secondLargest

if \_\_name\_\_ == "\_\_main\_\_":

arr = (input("Enter elements of the list separated by space: "))

result = findLargest(arr)

print("The second largest element in the list is:", result)

5. Explain what indentation means in Python.

The primary purpose of indentation in Python is to define the scope of statements, such as those within loops, conditionals, functions, and classes. Consistent and proper indentation is crucial for the interpreter to understand the logical structure of the code. Indentation is not just a matter of style or convention in Python.

Python’s default indentation spaces are four spaces. The number of spaces, however, is entirely up to the user. However, a minimum of one space is required to indent a statement.

Indentation is not permitted on the first line of Python code.

6. Write a program to perform set difference operation.

def setdifference(set1, set2):

return set1 - set2

if \_\_name\_\_ == "\_\_main\_\_":

set1 = set(input("Enter elements of set 1:”))

set2 = set(input("Enter elements of set 2:”))

difference = setdifference(set1, set2)

print(“Difference of two sets :” +difference)

7.Write a Python program to print numbers from 1 to 10 using a while loop.

num = 1

while num <= 10:

print(num)

num += 1

8. Write a program to calculate the factorial of a number using a while loop.

def factorial(n):

if n == 0 or n == 1:

return 1

fact = 1

while n > 1:

fact \*= n

n -= 1

return fact

if \_\_name\_\_ == "\_\_main\_\_":

num = int(input("Enter a number to calculate its factorial: "))

if num < 0:

print("Factorial is not defined for negative numbers.")

else:

result = factorial(num)

print(result)

9. Write a Python program to check if a number is positive, negative, or zero using if-elif-else statements.

num = 10

If num > = 0:

If num == 0:

Print(‘0’)

else:

print(“Positive”)

else:

print(“Negative”)

10. Write a program to determine the largest among three numbers using conditional statements.

int main() {

double n1, n2, n3;

printf("Enter three different numbers: ");

scanf("%lf %lf %lf", &n1, &n2, &n3);

if (n1 >= n2 && n1 >= n3)

printf(“Largest number”, n1);

if (n2 >= n1 && n2 >= n3)

printf(“largest number", n2);

if (n3 >= n1 && n3 >= n2)

printf("Largest number", n3);

return 0;

}

11. Write a Python program to create a numpy array filled with ones of given shape.

import numpy as np

def createonesarray(shape):

onesarray = np.ones(shape)

return ones\_array

print(“Array “(shape, onesarray))

12. Write a program to create a 2D numpy array initialized with random integers.

import numpy as np

rows = int(input("Enter rows: "))

cols = int(input("Enter columns: "))

randomarray = np.random.randint(0, 100, size=(rows, cols))

print("Random 2D array:")

print(randomarray)

13. Write a Python program to generate an array of evenly spaced numbers over a specified range using linspace.

import numpy as np

start = 4

stop = 16

num\_elements = 20

evenly\_spaced\_array = np.linspace(start, stop, num\_elements)

print("Array of evenly spaced numbers:")

print(evenly\_spaced\_array)

14. Write a program to generate an array of 10 equally spaced values between 1 and 100 using linspace.

import numpy as np

equally\_spaced\_array = np.linspace(1, 100, 10)

print("Array of 10 equally spaced values between 1 and 100:")

print(equally\_spaced\_array)

15. Write a Python program to create an array containing even numbers from 2 to 20 using arange.

import numpy as np

even\_numbers\_array = np.arange(2, 21, 2)

print("Array of even numbers from 2 to 20:")

print(even\_numbers\_array)

16. Write a program to create an array containing numbers from 1 to 10 with a step size of 0.5 using arrange.

import numpy as np

numbers\_array = np.arange(1, 10.5, 0.5)

print("Array of numbers from 1 to 10 with a step size of 0.5:")

print(numbers\_array)