

## ASSESSMENT -1

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

```
# Function to calculate the area of a rectangle
```

```
def rectangle_area(length, width):
```

```
    # Validate the input parameters
```

```
    if not isinstance(length, (int, float)) or length <= 0:
```

```
        raise ValueError("Length must be a positive number.")
```

```
    if not isinstance(width, (int, float)) or width <= 0:
```

```
        raise ValueError("Width must be a positive number.")
```

```
    # Calculate the area
```

```
    area = length * width
```

```
    return area
```

```
# Example usage
```

```
length = 5
```

```
width = 3
```

```
area = rectangle_area(length, width)
```

```
print(f"The area of the rectangle with length {length} and width {width} is {area}.")
```

2. Write a program to convert miles to kilometers

```
# Function to convert miles to kilometers
```

```
def miles_to_km(miles):
```

```
    # Define the conversion factor
```

```
    CONVERSION_FACTOR = 1.60934
```

```
    # Validate the input parameter
```

```
    if not isinstance(miles, (int, float)):
```

```
        raise ValueError("Miles must be a number.")
```

```
    # Convert miles to kilometers
```

```
km = miles * CONVERSION_FACTOR
```

```
return km
```

```
# Example usage
```

```
miles = 10
```

```
km = miles_to_km(miles)
```

```
print(f'{miles} miles is equal to {km} kilometers.")
```

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

```
def is_palindrome(s):
```

```
    """
```

```
    Check if a given string is a palindrome.
```

```
    :param s: The string to check.
```

```
    :return: True if the string is a palindrome, False otherwise.
```

```
    """
```

```
    # Remove any non-alphanumeric characters and convert to lowercase
```

```
    s = "".join(c.lower() for c in s if c.isalnum())
```

```
    # Check if the string is equal to its reverse
```

```
    return s == s[::-1]
```

```
# Example usage
```

```
s = "racecar"
```

```
if is_palindrome(s):
```

```
    print(f'{s} is a palindrome.")
```

```
else:
```

```
    print(f'{s} is not a palindrome.")
```

```
s = "hello"
```

```
if is_palindrome(s):
```

```
    print(f'{s} is a palindrome.")
```

```
else:
```

```
    print(f'{s} is not a palindrome.")
```

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

```

def second_largest(numbers):
    """
    Find the second largest element in a list of numbers.

    :param numbers: The list of numbers.
    :return: The second largest element in the list.
    """
    # Check if the list is empty
    if not numbers:
        raise ValueError("List cannot be empty.")

    # Initialize variables to store the largest and second largest elements
    largest = numbers[0]
    second_largest = None

    # Iterate over the list to find the largest and second largest elements
    for num in numbers:
        # If the current number is larger than the largest element, update the largest and second largest
        # elements
        if num > largest:
            second_largest = largest
            largest = num

        # If the current number is between the largest and second largest elements, update the second
        # largest element
        elif second_largest is None or num > second_largest:
            second_largest = num

    # Check if there are at least two distinct elements in the list
    if second_largest is None:
        raise ValueError("List must contain at least two distinct elements.")

    # Return the second largest element
    return second_largest

# Example usage

```

```
numbers = [1, 2, 3, 4, 5]
second_largest_num = second_largest(numbers)
print(f"The second largest element in the list is {second_largest_num}.")
numbers = [5, 2, 9, 2, 7]
second_largest_num = second_largest(numbers)
print(f"The second largest element in the list is {second_largest_num}.")
```

#### 5. Explain what indentation means in Python

Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code

indentation is used to define the scope and structure of the code. It is used to indicate which statements belong to which blocks of code, such as the body of a function, loop, or conditional statement.

#### 6. Write a program to perform set difference operation

```
# Define two sets
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}

# Perform set difference operation
set_difference = set1.difference(set2)

# Print the result
print("Set difference of set1 and set2 is:", set_difference)

# Perform set difference operation using the - operator
set_difference2 = set1 - set2

# Print the result
print("Set difference of set1 and set2 using - operator is:", set_difference2)
```

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

```
# Initialize a counter variable to 1
counter = 1

# Use a while loop to print numbers from 1 to 10
```

```
while counter <= 10:
```

```
    print(counter)
```

```
    counter += 1
```

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

```
# Get the number from the user
```

```
number = int(input("Enter a number: "))
```

```
# Initialize the factorial variable to 1
```

```
factorial = 1
```

```
# Use a while loop to calculate the factorial
```

```
while number > 1:
```

```
    factorial *= number
```

```
    number -= 1
```

```
# Print the result
```

```
print("The factorial of", number, "is", factorial)
```

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

```
# Get the number from the user
```

```
number = float(input("Enter a number: "))
```

```
# Check if the number is positive, negative, or zero
```

```
if number > 0:
```

```
    print("The number is positive.")
```

```
elif number < 0:
```

```
    print("The number is negative.")
```

```
else:
```

```
    print("The number is zero.")
```

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

```
# Get the three numbers from the user
```

```
num1 = float(input("Enter the first number: "))
```

```

num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))

# Determine the largest among the three numbers
if num1 >= num2 and num1 >= num3:
    largest = num1
elif num2 >= num1 and num2 >= num3:
    largest = num2
else:
    largest = num3

# Print the result
print("The largest number is", largest)

```

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

```

import numpy as np

# Get the shape of the array from the user
rows = int(input("Enter the number of rows: "))
cols = int(input("Enter the number of columns: "))

# Create a NumPy array filled with ones of the given shape
arr = np.ones((rows, cols))

# Print the resulting array
print("The resulting array is:")
print(arr)

```

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

```

import numpy as np

# Get the shape of the array from the user
rows = int(input("Enter the number of rows: "))
cols = int(input("Enter the number of columns: "))

# Create a NumPy array initialized with random integers of the given shape

```

```
arr = np.random.randint(low=-10, high=10, size=(rows, cols))
```

```
# Print the resulting array
```

```
print("The resulting array is:")
```

```
print(arr)
```

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

```
import numpy as np
```

```
# Get the input parameters from the user
```

```
start = float(input("Enter the start of the range: "))
```

```
stop = float(input("Enter the end of the range: "))
```

```
num = int(input("Enter the number of elements: "))
```

```
# Generate an array of evenly spaced numbers over the specified range
```

```
arr = np.linspace(start, stop, num)
```

```
# Print the resulting array
```

```
print("The resulting array is:")
```

```
print(arr)
```

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

```
import numpy as np
```

```
# Generate an array of 10 equally spaced values between 1 and 100
```

```
arr = np.linspace(1, 100, 10)
```

```
# Print the resulting array
```

```
print("The resulting array is:")
```

```
print(arr)
```

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

```
import numpy as np
```

```
# Create an array containing even numbers from 2 to 20 using arrange
```

```
arr = np.arange(2, 21, 2)
```

```
# Print the resulting array
```

```
print("The resulting array is:")
```

```
print(arr)
```

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
```

```
# Create an array containing numbers from 1 to 10 with a step size of 0.5 using arrange
```

```
arr = np.arange(1, 10.5, 0.5)
```

```
# Print the resulting array
```

```
print("The resulting array is:")
```

```
print(arr)
```