ASSESSMENT - 1

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

Ans :
```
def calculate_rectangle_area(length, width):
    return length * width


def main():
    length = float(input("Enter the length of the rectangle: "))
    width = float(input("Enter the width of the rectangle: "))


    area = calculate_rectangle_area(length, width)
    print("The area of the rectangle is:", area)


if __name__ == "__main__":
    main()
```

2. Write a program to convert miles to kilometers

Ans :
```
def miles_to_kilometers(miles):
    return miles * 1.60934


def main():
    miles = float(input("Enter the distance in miles: "))
    kilometers = miles_to_kilometers(miles)
    print(f"{miles} miles is equal to {kilometers} kilometers.")


if __name__ == "__main__":
    main()
```

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

Ans :
```
def is_palindrome(s):
    # Remove spaces and convert to lowercase for case-insensitive comparison
    s = s.replace(" ", "").lower()
    # Compare the string with its reverse
    return s == s[::-1]

# Test the function
def main():
    string = input("Enter a string: ")
    if is_palindrome(string):
        print("The string is a palindrome.")
    else:
        print("The string is not a palindrome.")

if __name__ == "__main__":
    main()
```

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

Ans :
```
def find_second_largest(numbers):
    if len(numbers) < 2:
        return "List must contain at least two elements"

    largest = float('-inf')
    second_largest = float('-inf')

    for num in numbers:
        if num > largest:
            second_largest = largest
            largest = num
        elif num > second_largest and num != largest:
            second_largest = num

    return second_largest

# Test the function
```

```
def main():
    numbers = [int(x) for x in input("Enter the list of numbers separated by space: ").split()]
    second_largest = find_second_largest(numbers)
    print("The second largest element in the list is:", second_largest)


if __name__ == "__main__":
    main()
```

5. Explain what indentation means in Python

Ans : Indentation refers to the spaces or tabs placed at the beginning of a line of code to define a block of code. Unlike many other programming languages where indentation is simply for readability, in Python, indentation is significant and serves as a fundamental part of the language's syntax.

Indentation is used to denote the beginning and end of control flow structures such as loops, conditional statements, and function definitions. It helps Python to understand the structure of the code and identify which statements belong to which block.

For example, consider this Python code snippet:

```
if x > 0:
    print("x is positive")
    print("This line is inside the if block")
print("This line is outside the if block")
```

6. Write a program to perform set difference operation.

Ans :
```
def set_difference(set1, set2):
    return set1 - set2


def main():
    set1 = set(input("Enter the elements of the first set separated by space: ").split())
    set2 = set(input("Enter the elements of the second set separated by space: ").split())


    difference = set_difference(set1, set2)
    print("The set difference (set1 - set2) is:", difference)
```

```
if __name__ == "__main__":
    main()
```

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

Ans :
```
def print_numbers():
    num = 1
    while num <= 10:
        print(num)
        num += 1

# Call the function to print numbers from 1 to 10
print_numbers()
```

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

Ans :
```
def calculate_factorial(n):
    factorial = 1
    while n > 1:
        factorial *= n
        n -= 1
    return factorial

def main():
    number = int(input("Enter a number to calculate its factorial: "))
    if number < 0:
        print("Factorial is not defined for negative numbers.")
    else:
        factorial = calculate_factorial(number)
        print(f"The factorial of {number} is: {factorial}")

if __name__ == "__main__":
    main()
```

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

Ans :
```
def check_number(n):
    if n > 0:
        print("The number is positive.")
    elif n < 0:
        print("The number is negative.")
    else:
        print("The number is zero.")

def main():
    number = float(input("Enter a number: "))
    check_number(number)

if __name__ == "__main__":
    main()
```

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

Ans :
```
def find_largest(num1, num2, num3):
    if num1 >= num2 and num1 >= num3:
        return num1
    elif num2 >= num1 and num2 >= num3:
        return num2
    else:
        return num3

def main():
    num1 = float(input("Enter the first number: "))
    num2 = float(input("Enter the second number: "))
    num3 = float(input("Enter the third number: "))
    largest = find_largest(num1, num2, num3)
    print("The largest number among", num1, ",", num2, ", and", num3, "is:", largest)

if __name__ == "__main__":
    main()
```

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

Ans : import numpy as np

```
def create_ones_array(shape):
    ones_array = np.ones(shape)
    return ones_array

def main():
    rows = int(input("Enter the number of rows: "))
    columns = int(input("Enter the number of columns: "))

    shape = (rows, columns)
    ones_array = create_ones_array(shape)

    print("Array of ones with shape", shape, "is:")
    print(ones_array)

if __name__ == "__main__":
    main()
```

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

Ans : import numpy as np

```
def create_random_array(rows, columns, low, high):
    random_array = np.random.randint(low, high, size=(rows, columns))
    return random_array

def main():
    rows = int(input("Enter the number of rows: "))
    columns = int(input("Enter the number of columns: "))
    low = int(input("Enter the lower bound for random integers: "))
    high = int(input("Enter the upper bound for random integers: "))

    random_array = create_random_array(rows, columns, low, high)

    print("2D NumPy array initialized with random integers is:")
    print(random_array)

if __name__ == "__main__":
    main()
```

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

Ans : import numpy as np

```
def generate_array(start, stop, num):
    evenly_spaced_array = np.linspace(start, stop, num)
    return evenly_spaced_array

def main():
    start = float(input("Enter the start value: "))
    stop = float(input("Enter the stop value: "))
    num = int(input("Enter the number of evenly spaced numbers: "))

    evenly_spaced_array = generate_array(start, stop, num)
    print("Array of evenly spaced numbers over the specified range is:")
    print(evenly_spaced_array)


if __name__ == "__main__":
    main()
```

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

Ans : import numpy as np

```
def generate_array():
    equally_spaced_array = np.linspace(1, 100, 10)
    return equally_spaced_array


def main():
    equally_spaced_array = generate_array()


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


if __name__ == "__main__":
    main()
```

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

Ans : import numpy as np

```
def create_even_array():
    even_array = np.arange(2, 21, 2)
    return even_array


def main():
    even_array = create_even_array()

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

if __name__ == "__main__":
    main()
```

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

Ans : import numpy as np

```
def create_array():
    array = np.arange(1, 11, 0.5)
    return array

def main():
    result_array = create_array()
    print("Array containing numbers from 1 to 10 with a step size of 0.5:")
    print(result_array)

if __name__ == "__main__":
    main()
```