Assessment-1

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

Ans.
```
def areaofRect(x, y):
     return x*y

    print("Enter Length and Breadth of Rectangle: ", end="")
   l = float(input())
   b = float(input())
   a = areaofRect(l, b)
   print("\nArea = ", a)
```

2. Write a program to convert miles to kilometers

Ans.
```
miles = float(input("Please enter miles:"))

   kilometers = miles * 1.6

   print(kilometers, " Kilometers")
```

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

Ans.
```
def isPalindrome(string):
     if (string == string[::-1]) :
            return "The string is a palindrome."
     else:
           return "The string is not a palindrome."

   string = input ("Enter string: ")
   print(isPalindrome(string))
```

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

Ans.
```
def second_largest(arr):

    if len(arr) < 2:

    return "List must have at least two elements"

    largest = second_largest = float('-inf')

    for num in arr:

      if num > largest:

        second_largest = largest

        largest = num
        elif num > second_largest and num != largest:
            second_largest = num
    if second_largest == float('-inf'):
        return "There is no second largest element"
    return second_largest
```

5. Explain what indentation means in python

Ans. Indentation is used to define the structure and hierarchy of code. It serves as a way to group blocks of code together. Unlike many other programming languages that use curly braces {} or keywords like begin and end to denote code blocks, Python uses indentation.

Here's what indentation signifies in Python:

Block Structure: Indentation is used to define blocks of code such as loops, conditional statements, function definitions, and class definitions. Each level of indentation represents a nested block.

Readability: Python places a strong emphasis on code readability. By enforcing indentation, Python code tends to be more readable and visually organized.

No Delimiters: In Python, there are no explicit block delimiters like curly braces {}. Instead, indentation is used to determine the start and end of blocks.

Consistency: Since Python requires consistent indentation, it enforces a consistent coding style across projects and among developers. This reduces the likelihood of syntax errors and improves code maintainability.

Whitespace: Python is sensitive to whitespace, so the spaces or tabs used for indentation must be consistent throughout the codebase. Mixing spaces and tabs can lead to indentation errors.

Here's an example to illustrate indentation in Python:

```
if True:
    print("This line is indented")
  if False:
      print("This line is indented further")
  print("Back to the first indentation level")
print("This line is not indented")
```

6.Write a program to perform set difference operation

Ans. The set difference operation in Python can be performed using the - operator or the difference() method. Below is a program demonstrating both methods:

```
# Using the '-' operator for set difference
def set_difference_operator(set1, set2):
   return set1 - set2
# Using the difference() method for set difference
def set_difference_method(set1, set2):
   return set1.difference(set2)
```

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

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

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

```
Ans.   def calculate_factorial(number):
       factorial_result = 1
       while number > 0:
       factorial_result *= number
       number -= 1
       return factorial_result
       user_input = int(input("Enter a number to calculate its factorial: "))
       result = calculate_factorial(user_input)
       print(f"The factorial of {user_input} is: {result}")
```

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

```
Ans.    number = float(input("Enter a number: "))
        if number > 0:
        print("The entered number is positive.")
        elif number < 0:
        print("The entered number is negative.")
        else:
```

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

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

```
Ans. num1 = float(input("Enter first number: "))
     num2 = float(input("Enter second number: "))
     num3 = float(input("Enter third number: "))
     if num1 >= num2 and num1 >= num3:
         largest = num1
     elif num2 >= num1 and num2 >= num3:
         largest = num2
     else:
         largest = num3
     print("The largest number is:", largest)
```

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
      # Example usage
      shape = (3, 4)  # Shape of the array (3 rows, 4 columns)
      ones_array = create_ones_array(shape)
      print("Array filled with ones of shape", shape, ":\n", ones_array)
```

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

```
Ans.  import numpy as np
      rows = int(input("Enter the number of rows: "))
      cols = int(input("Enter the number of columns: "))
      random_array = np.random.randint(low=0, high=100, size=(rows, cols))
      print("2D Array filled with random integers:")
      print(random_array)
```

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

Ans.
```
import numpy as np

    # Define the range and the number of points

    start = 0

    stop = 10

    num_points = 5

    # Generate the array of evenly spaced numbers

    result_array = np.linspace(start, stop, num_points)

    # Print the result

    print("Array of evenly spaced numbers:")

    print(result_array)
```

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

Ans.
```
import numpy as np

    # Define the range and the number of points

    start = 1

    stop = 100

    num_points = 10

    # Generate the array of evenly spaced numbers

    result_array = np.linspace(start, stop, num_points)

    # Print the result

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

    print(result_array)
```

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

Ans.
```
import numpy as np
    even_numbers = np.arange(2, 21, 2)
    print(even_numbers)
```

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

Ans.
```
import numpy as np
    numbers = np.arange(1, 10.5, 0.5)
    print(numbers)
```