

Assessment-1

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

```
def calculate_rectangle_area(length, width):  
    area = length * width  
    return area  
  
# Manually set values for testing  
  
length = 5.0  
width = 3.0  
  
# Calculate and print the area  
area = calculate_rectangle_area(length, width)  
print(f"The area of the rectangle with length {length} and width {width} is: {area}")
```

Output:

```
The area of the rectangle with length 5.0 and width 3.0 is: 15.0
```

2. Write a program to convert miles to kilometers

```
def miles_to_kilometers(miles):  
    # Conversion factor: 1 mile is approximately 1.60934 kilometers  
    kilometers = miles * 1.60934  
    return kilometers  
  
# Manually set the value for testing  
  
miles = 5.0  
  
# Convert miles to kilometers and print the result  
kilometers = miles_to_kilometers(miles)  
print(f"{miles} miles is equal to {kilometers:.2f} kilometers")
```

Output:

```
5.0 miles is equal to 8.05 kilometers
```

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

```
def is_palindrome(s):
    cleaned_string = ''.join(char.lower() for char in s if char.isalnum())
    return cleaned_string == cleaned_string[::-1]

# Manually set the value for testing
input_string = "A man, a plan, a canal, Panama!"
result = is_palindrome(input_string)

if result:
    print(f"{input_string} is a palindrome.")
else:
    print(f"{input_string} is not a palindrome.")
```

Output:

A man, a plan, a canal, Panama! is a palindrome.

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

```
def second_largest_element(lst):
    if len(lst) < 2:
        return "List should have at least two elements."

    largest = second_largest = float('-inf')

    for num in lst:
        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."
else:
    return f"The second largest element is: {second_largest}"
```

Example usage:

```
my_list = [10, 5, 8, 20, 15]
result = second_largest_element(my_list)
print(result)
```

Output:

```
The second largest element is: 15
```

5.. Explain what indentation means in Python

```
# Define a value for x
x = 5

# Check if x is positive or non-positive
if x > 0:
    print("x is positive")
    print("This is inside the if block")
else:
    print("x is non-positive")
    print("This is inside the else block")
```

Output:

```
x is positive
This is inside the if block
```

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

```
# Define two sets

set1 = {1, 2, 3, 4, 5}

set2 = {3, 4, 5, 6, 7}

# Using the - operator for set difference

difference_result_operator = set1 - set2

print("Set difference using - operator:", difference_result_operator)

# Using the difference() method for set difference
difference_result_method = set1.difference(set2)

print("Set difference using difference() method:", difference_result_method)
```

Output:

```
Set difference using - operator: {1, 2}
Set difference using difference() method: {1, 2}
```

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

```
# Initialize a variable

number = 1

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

while number <= 10:

    print(number)

    number += 1
```

Output:

```
1
2
3
4
5
6
7
8
9
10
```

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

Function to calculate factorial using a while loop

```
def calculate_factorial(number):  
    factorial = 1  
  
    # Check if the number is non-negative  
    if number < 0:  
        return "Factorial is undefined for negative numbers"  
  
    # Calculate factorial using a while loop  
    while number > 0:  
        factorial *= number  
        number -= 1  
  
    return factorial  
  
# Manually set the value for testing  
user_input = 5  
  
# Calculate and print the factorial  
result = calculate_factorial(user_input)  
print(f"The factorial of {user_input} is: {result}")
```

Output:

The factorial of 5 is: 120

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

Manually set the value for testing

```
number = 5

# 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.")
```

Output:

The number is positive.

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

```
# Predefined values for three numbers

num1 = 25
num2 = 14
num3 = 32

# 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(f"The largest number among {num1}, {num2}, and {num3} is: {largest}")
```

Output:

The largest number among 25, 14, and 32 is: 32

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

```
import numpy as np

# Specify the shape of the array

shape = (3, 4) # Example shape: 3 rows, 4 columns

# Create a NumPy array filled with ones

ones_array = np.ones(shape)

# Print the resulting array

print("NumPy array filled with ones:")

print(ones_array)
```

Output:

```
NumPy array filled with ones:
[[1.  1.  1.  1.]
 [1.  1.  1.  1.]
 [1.  1.  1.  1.]]
```

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

```
import numpy as np

# Specify the shape of the array

rows = 3

columns = 4

# Create a 2D NumPy array initialized with random integers

random_int_array = np.random.randint(low=1, high=100, size=(rows, columns))# Print the resulting array

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

print(random_int_array)
```

Output:

```
2D NumPy array initialized with random integers:
[[34 82 47 59]]
```

```
[74 67 80 96]
[ 8 56 82  5]]
```

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

```
import numpy as np

# Specify the range and the number of elements

start = 0

stop = 10

num_elements = 5

# Generate an array of evenly spaced numbers using linspace

evenly_spaced_array = np.linspace(start, stop, num_elements)

# Print the resulting array

print("Array of evenly spaced numbers:")

print(evenly_spaced_array)
```

Output:

```
Array of evenly spaced numbers:
[ 0.  2.5  5.  7.5 10. ]
```

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

equally_spaced_array = np.linspace(1, 100, 10)

# Print the resulting array

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

Output:

Array of 10 equally spaced values between 1 and 100:
[1. 12. 23. 34. 45. 56. 67. 78. 89. 100.]

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

```
import numpy as np

# Create an array containing even numbers from 2 to 20 using arange
even_numbers_array = np.arange(2, 21, 2)

# Print the resulting array
print("Array containing even numbers from 2 to 20:")
print(even_numbers_array)
```

Output:

Array containing even numbers from 2 to 20:
[2 4 6 8 10 12 14 16 18 20]

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

```
import numpy as np

# Create an array containing numbers from 1 to 10 with a step size of 0.5 using arange
array_with_step = np.arange(1, 11, 0.5)

# Print the resulting array
print("Array containing numbers from 1 to 10 with a step size of 0.5:")
print(array_with_step)
```

Output:

Array containing numbers from 1 to 10 with a step size of 0.5:
[1. 1.5 2. 2.5 3. 3.5 4. 4.5 5. 5.5 6. 6.5 7. 7.5
8. 8.5 9. 9.5 10. 10.5]