

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):  
    area = length * width  
  
    return area  
  
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)
```

2. Write a program to convert miles to kilometers.

Ans.

```
miles = float(input("Enter the distance in miles: "))  
  
kilometers = miles * 1.60934  
  
print("The distance in kilometers is:", kilometers)
```

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

Ans.

```
def is_palindrome(s):  
    return s == s[::-1]  
  
input_string = input("Enter a string: ")  
  
if is_palindrome(input_string):  
    print("The string is a palindrome.")  
else:  
    print("The string is not a palindrome.")
```

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

Ans.

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

5. Explain what indentation means in python.

Ans. In Python, indentation is used to define the structure and hierarchy of code blocks. Unlike many other programming languages that use braces `{}` or keywords like `begin` and `end` to denote blocks of code, Python uses indentation to indicate the beginning and end of blocks.

Here's how indentation works in Python:

1. Indentation level: Indentation is typically done using spaces or tabs at the beginning of lines. The number of spaces or tabs used for indentation should be consistent throughout the code.
2. Code blocks: Blocks of code are defined by indentation. Statements that are at the same level of indentation are considered part of the same block. Blocks can contain one or more statements.
3. Hierarchy: Nested blocks are created by increasing the level of indentation. This creates a hierarchical structure where inner blocks are contained within outer blocks.
4. Colon `:` : In Python, the colon (`:`) is used to indicate the start of an indented code block, such as in control flow statements (`if`, `else`, `for`, `while`, etc.) and function or class definitions.

Here's an example to illustrate indentation in Python:

if condition:

```
    print("Condition is true")  
    print("This statement is also part of the block")  
print("This statement is outside the block")
```

6. Write a program to perform set difference operation.

Ans.

```
set1 = {1, 2, 3, 4, 5}  
set2 = {3, 4, 5, 6, 7}  
difference = set1 - set2  
print("Set difference (set1 - set2):", difference)
```

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
```

In simpler terms:

- We start with the number 1.
- We keep printing the current number as long as it's less than or equal to 10.

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

Ans.

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

factorial = 1

current = 1

while current <= num:

    factorial *= current

    current += 1

print("Factorial of", num, "is:", factorial)
```

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("Positive")

elif number < 0:

    print("Negative")

else:

    print("Zero")
```

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

Ans.

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

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

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

if num1 >= num2 and num1 >= num3:

    print("The largest number is:", num1)

elif num2 >= num1 and num2 >= num3:

    print("The largest number is:", num2)

else:

    print("The largest number is:", num3)
```

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

Ans.

```
import numpy as np

shape = input("Enter the shape of the array (e.g., '3 4' for a 3x4 array): ")

shape_tuple = tuple(map(int, shape.split()))

ones_array = np.ones(shape_tuple)

print("Array filled with ones of shape", shape_tuple, ":\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(1, 100, size=(rows, cols))

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

# Generate an array of 10 evenly spaced numbers between 1 and 10
result_array = np.linspace(1, 10, 10)

# Print the resulting array
print("Array of evenly spaced numbers over the specified range:")

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

# Generate an array of 10 equally spaced values between 1 and 100
result_array = np.linspace(1, 100, 10)

# Print the resulting array
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

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

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

```
# Print the resulting array
```

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

```
print(even_array)
```

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

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

```
array_with_step = np.arange(1, 10.5, 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)
```