

AI / ML Training

Assessment

1. Write a Python program to calculate the area of a rectangle given its length and width
2. Write a program to convert miles to kilometers
3. Write a function to check if a given string is a palindrome.
4. Write a Python program to find the second largest element in a list.
5. Explain what indentation means in Python.
6. Write a program to perform set difference operation.
7. Write a Python program to print numbers from 1 to 10 using a while loop.
8. Write a program to calculate the factorial of a number using a while loop.
9. Write a Python program to check if a number is positive, negative, or zero using if-elif-else statements.
10. Write a program to determine the largest among three numbers using conditional statements.
11. Write a Python program to create a numpy array filled with ones of given shape.
12. Write a program to create a 2D numpy array initialized with random integers.
13. Write a Python program to generate an array of evenly spaced numbers over a specified range using linspace.
14. Write a program to generate an array of 10 equally spaced values between 1 and 100 using linspace.
15. Write a Python program to create an array containing even numbers from 2 to 20 using arange.
16. Write a program to create an array containing numbers from 1 to 10 with a step size of 0.5 using arange.

1.

```
l = float(input('Enter Length of the Rectangle: '))
```

```
b = float(input('Enter Breadth of the Rectangle: '))
```

```
area = l * b
```

```
print(f"Area of rectangle is {area:.2f}")
```

2.

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

```
    km = miles * 1.60934
```

```
    return km
```

```
# Input miles from user
```

```
miles = float(input("Enter miles: "))
```

```
# Convert miles to kilometers
```

```
kilometers = miles_to_km(miles)
```

```
# Display the result
```

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

3.

```
def is_palindrome(s):  
    s = s.lower() # Convert string to lowercase  
    s = ''.join(filter(str.isalnum, s)) # Remove non-alphanumeric characters  
    return s == s[::-1] # Check if the string is equal to its reverse  
  
# Test the function  
print(is_palindrome("A man, a plan, a canal, Panama!")) # True  
print(is_palindrome("race a car")) # False
```

4.

```
def second_largest(numbers):  
    # Initialize the two largest variables  
    largest = second_largest = float('-inf')  
  
    # Iterate through the list  
    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  
numbers = [10, 20, 4, 45, 99]  
print("Second largest element is:", second_largest(numbers))
```

5.

In Python, indentation is used to define the structure and hierarchy of the code. It is crucial for Python's syntax and is used to group statements together. Here's what indentation means in Python:

**Block of Code:** Indentation is used to define a block of code, such as the body of a loop, a conditional statement, or a function. All statements with the same level of indentation are considered part of the same block.

**Hierarchy:** Indentation is used to show the hierarchical relationship between different blocks of code. For example, statements inside a loop are indented to show that they are part of the loop body.

**Readability:** Indentation improves the readability of the code by visually indicating the structure of the program. It makes it easier for programmers to understand the flow of the code.

**No Braces:** Unlike many other programming languages, Python does not use braces ({} ) to define blocks of code. Instead, it uses indentation to determine the beginning and end of blocks.

**Consistency:** Python requires consistent indentation throughout the code. Mixing tabs and spaces for indentation is not allowed and will result in an error.

Here's an example to illustrate the use of indentation in Python:

```
if x > 0:
    print("x is positive") # This statement is indented and part of the if block
    if x % 2 == 0:
        print("x is even") # This statement is indented and part of the inner if block
else:
    print("x is non-positive") # This statement is not indented and is not part of the if block

6.
set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 5, 6, 7}

# Perform set difference operation
result_set = set1 - set2

# Print the result
```

```
print("Set Difference:", result_set)
```

7.

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

8.

```
num = int(input("Enter a number: "))
factorial = 1

if num < 0:
    print("Factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    while num > 0:
        factorial *= num
        num -= 1
    print("The factorial of", num, "is", factorial)
```

9.

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

if num > 0:
    print("The number is positive")
elif num == 0:
    print("The number is zero")
else:
```

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

10.

```
import numpy as np
```

```
# Create a 3x4 array filled with ones
```

```
shape = (3, 4)
```

```
arr = np.ones(shape)
```

```
print("Array filled with ones:")
```

```
print(arr)
```

11.

```
import numpy as np
```

```
# Specify the shape of the array
```

```
rows = 3
```

```
cols = 4
```

```
# Create a 2D array with random integers between 0 and 9
```

```
arr = np.random.randint(0, 10, (rows, cols))
```

```
print("2D Array initialized with random integers:")
```

```
print(arr)
```

12.

```
import numpy as np
```

```
# Generate an array of 5 evenly spaced numbers between 0 and 10
```

```
arr = np.linspace(0, 10, 5)
```

```
print("Array of evenly spaced numbers:")  
print(arr)
```

13.

```
import numpy as np
```

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

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

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

```
print(arr)import numpy as np
```

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

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

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

```
print(arr)
```

14.

```
import numpy as np
```

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

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

```
print(arr)
```

15.

```
import numpy as np
```

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

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

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

```
print(arr)
```

16.

```
import numpy as np
```

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

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

```
print("Array containing numbers from 1 to 10 with a step size of 0.5:")
```

```
print(arr)
```