Basic Level:

1. Write a Python program to print the numbers from 1 to 10 using a `for` loop.

```
for i in range(1, 11):
   print(i)
```

2. Create a program that calculates the sum of all numbers in a list using a `for` loop.

```
numbers = [1, 2, 3, 4, 5]
sum = 0
for num in numbers:
   sum += num
print("Sum:", sum)
```

3. Write a program to print the characters of a string in reverse order using a `for` loop.

```
string = "hello"
for char in reversed(string):
   print(char)
```

4. Develop a program that finds the factorial of a given number using a `for` loop.

```
number = 5
factorial = 1
for i in range(1, number + 1):
   factorial *= i
print("Factorial of", number, ":", factorial)
```

5. Create a program to print the multiplication table of a given number using a `for` loop.

```
number = 5
for i in range(1, 11):
   print(number, "x", i, "=", number * i)
```

6. Write a program that counts the number of even and odd numbers in a list using a `for` loop.

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_count = 0
odd_count = 0
for num in numbers:
   if num % 2 == 0:
      even_count += 1
   else:
      odd_count += 1
print("Number of even numbers:", even_count)
print("Number of odd numbers:", odd_count)
```

7. Develop a program that prints the squares of numbers from 1 to 5 using a `for` loop.

```
for i in range(1, 6):
    print("Square of", i, ":", i ** 2)
```

8. Create a program to find the length of a string without using the `len()` function.

```
string = "hello"
length = 0
for char in string:
    length += 1
print("Length of the string:", length)
```

9. Write a program that calculates the average of a list of numbers using a `for` loop.

```
numbers = [1, 2, 3, 4, 5]
sum = 0
for num in numbers:
    sum += num
average = sum / len(numbers)
print("Average:", average)
```

10. Develop a program that prints the first `n` Fibonacci numbers using a `for` loop.
Intermediate Level:

```
n = 10
fibonacci = [0, 1]
for i in range(2, n):
    fibonacci.append(fibonacci[i - 1] + fibonacci[i - 2])
print("First", n, "Fibonacci numbers:", fibonacci)
```

10. Write a program to check if a given list contains any duplicates using a `for` loop.

```
numbers = [1, 2, 3, 4, 5, 5, 6, 7, 8, 9]
for i in range(len(numbers)):
    for j in range(i + 1, len(numbers)):
        if numbers[i] == numbers[j]:
            print("List contains duplicates")
            break
```

11. Create a program that prints the prime numbers in a given range using a `for` loop.

```
start = 10
end = 50
for num in range(start, end + 1):
    if num > 1:
        is_prime = True
        for i in range(2, int(num ** 0.5) + 1):
            if num % i == 0:
                is_prime = False
                break
        if is_prime:
            print(num)
```

12. Develop a program that counts the number of vowels in a string using a `for` loop.

```
string = "hello world"
vowels = 'aeiou'
count = 0
for char in string:
    if char.lower() in vowels:
        count += 1
print("Number of vowels:", count)
```

13. Write a program to find the maximum element in a 2D list using a nested `for` loop.

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
max_element = matrix[0][0]
for row in matrix:
    for elem in row:
        if elem > max_element:
            max_element = elem
print("Maximum element:", max_element)
```

14. Create a program that removes all occurrences of a specific element from a list using a `for` loop.

```
numbers = [1, 2, 3, 4, 5, 2, 6, 7, 2, 8]
element_to_remove = 2
result = []
for num in numbers:
    if num != element_to_remove:
        result.append(num)
print("List after removal:", result)
```

15. Develop a program that generates a multiplication table for numbers from 1 to 5 using a nested `for` loop.

```
for i in range(1, 6):
    print("Multiplication table for", i)
    for j in range(1, 11):
        print(i, "x", j, "=", i * j)
```

16. Write a program that converts a list of Fahrenheit temperatures to Celsius using a `for` loop.

```
fahrenheit_temperatures = [32, 50, 68, 86, 104]
celsius_temperatures = []
for temp in fahrenheit_temperatures:
    celsius = (temp - 32) * 5/9
    celsius_temperatures.append(celsius)
print("Celsius temperatures:", celsius_temperatures)
```

17. Create a program to print the common elements from two lists using a `for` loop.

```
list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]
common_elements = []
for num in list1:
    if num in list2:
        common_elements.append(num)
print("Common elements:", common_elements)
```

19. Develop a program that prints the pattern of right-angled triangles using a `for` loop. Use ‘*’ to draw the
Pattern

```
for i in range(1, 6):
    print('*' * i)
```

20. Write a program to find the greatest common divisor (GCD) of two numbers using a `for` loop.
Advanced Level:

```
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

num1 = 48
num2 = 60
print("GCD of", num1, "and", num2, ":", gcd(num1, num2))
```

21. Create a program that calculates the sum of the digits of numbers in a list using a list comprehension.

```
numbers = [123, 456, 789]
sum_of_digits = [sum(int(digit) for digit in str(num)) for num in numbers]
print("Sum of digits of numbers:", sum_of_digits)
```

22. Write a program to find the prime factors of a given number using a `for` loop and list comprehension.

```
def prime_factors(n):
    factors = []
    for i in range(2, n+1):
        while n % i == 0:
            factors.append(i)
            n //= i
    return factors

number = 84
print("Prime factors of", number, ":", prime_factors(number))
```

23. Develop a program that extracts unique elements from a list and stores them in a new list using a list
comprehension.

```
numbers = [1, 2, 3, 4, 4, 5, 6, 6, 7]
unique_numbers = list(set(numbers))
print("Unique elements:", unique_numbers)
```

24. Create a program that generates a list of all palindromic numbers up to a specified limit using a list comprehension.

```
limit = 100
palindromic_numbers = [num for num in range(limit + 1) if str(num) == str(num)[::-1]]
print("Palindromic numbers up to", limit, ":", palindromic_numbers)
```

25. Write a program to flatten a nested list using list comprehension.

```
nested_list = [[1, 2, 3], [4, 5], [6, 7, 8]]
flattened_list = [num for sublist in nested_list for num in sublist]
print("Flattened list:", flattened_list)
```

26. Develop a program that computes the sum of even and odd numbers in a list separately using list comprehension.

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_sum = sum(num for num in numbers if num % 2 == 0)
odd_sum = sum(num for num in numbers if num % 2 != 0)
print("Sum of even numbers:", even_sum)
print("Sum of odd numbers:", odd_sum)
```

27. Create a program that generates a list of squares of odd numbers between 1 and 10 using list comprehension.

```
squares_of_odd = [x**2 for x in range(1, 11) if x % 2 != 0]
print("Squares of odd numbers:", squares_of_odd)
```

28. Write a program that combines two lists into a dictionary using list comprehension.

```
keys = ['a', 'b', 'c']
values = [1, 2, 3]
combined_dict = {key: value for key, value in zip(keys, values)}
print("Combined dictionary:", combined_dict)
```

29. Develop a program that extracts the vowels from a string and stores them in a list using list comprehension.

```
string = "hello world"
vowels = [char for char in string if char.lower() in 'aeiou']
print("Vowels:", vowels)
```

30. Create a program that removes all non-numeric characters from a list of strings using list comprehension.

```
strings = ['hello', '123', 'world', '456']
numeric_characters = [char for string in strings for char in string if char.isdigit()]
print("Numeric characters:", numeric_characters)
```

Challenge Level:

31. Write a program to generate a list of prime numbers using the Sieve of Eratosthenes algorithm and list comprehension.

```
limit = 50
primes = [i for i in range(2, limit + 1) if all(i % j != 0 for j in range(2, int(i**0.5) + 1))]
print("Prime numbers up to", limit, ":", primes)
```

32. Create a program that generates a list of all Pythagorean triplets up to a specified limit using list comprehension.

```
limit = 10
triplets = [(a, b, c) for a in range(1, limit) for b in range(a, limit) for c in range(b, limit) if a**2 + b**2 == c**2]
print("Pythagorean triplets up to", limit, ":", triplets)
```

Assignment Questions

Full Stack Data Science Pro

33. Develop a program that generates a list of all possible combinations of two lists using list comprehension.

```
list1 = [1, 2, 3]
list2 = ['a', 'b', 'c']
combinations = [(x, y) for x in list1 for y in list2]
print("All possible combinations:", combinations)
```

34. Write a program that calculates the mean, median, and mode of a list of numbers using list comprehension.

```
from statistics import mean, median, mode

numbers = [1, 2, 3, 4, 5, 5, 6, 7, 8, 9]
mean_value = mean(numbers)
median_value = median(numbers)
mode_value = mode(numbers)

print("Mean:", mean_value)
print("Median:", median_value)
print("Mode:", mode_value)
```

35. Create a program that generates Pascal's triangle up to a specified number of rows using list comprehension.

```
def generate_next_row(prev_row):
    return [1] + [prev_row[i] + prev_row[i+1] for i in range(len(prev_row) - 1)] + [1]

def generate_pascals_triangle(rows):
    triangle = [[1]]
    for _ in range(rows - 1):
        triangle.append(generate_next_row(triangle[-1]))
    return triangle
```

```
rows = 5
pascals_triangle = generate_pascals_triangle(rows)
print("Pascal's triangle:")
for row in pascals_triangle:
    print(row)
```

36. Develop a program that calculates the sum of the digits of a factorial of numbers from 1 to 5 using
list
comprehension.

```
factorial_digits_sum = [sum(int(digit) for digit in str(math.factorial(num))) for num in range(1, 6)]
print("Sum of digits of factorials:", factorial_digits_sum)
```

37. Write a program that finds the longest word in a sentence using list comprehension.

```
sentence = "This is a sentence to find the longest word"
longest_word = max(word for word in sentence.split(' '), key=len)
print("Longest word in the sentence:", longest_word)
```

38. Create a program that filters a list of strings to include only those with more than three vowels
using list
comprehension.

```
strings = ["hello", "world", "programming", "python", "algorithm"]
filtered_strings = [string for string in strings if sum(1 for char in string if char.lower() in 'aeiou') > 3]
print("Strings with more than three vowels:", filtered_strings)
```

39. Develop a program that calculates the sum of the digits of numbers from 1 to 1000 using list
comprehension.

```
digit_sum = [sum(int(digit) for digit in str(num)) for num in range(1, 1001)]
print("Sum of digits of numbers from 1 to 1000:", digit_sum)
```

40. Write a program that generates a list of prime palindromic numbers using list comprehension.

```
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True

prime_palindromic_numbers = [num for num in range(1, 1001) if is_prime(num) and str(num) ==
str(num)[::-1]]
print("Prime palindromic numbers:", prime_palindromic_numbers)
```