Amit Divekar | Assignment 2: Set B

Functions, Iterators and Generators - Set B

Q1. Create a function check_pass_fail(marks) that returns "Pass" if marks ≥ 40, else "Fail".

In [1]:

```
def check_pass_fail(marks):
    if marks >= 40:
        return "Pass"
    else:
        return "Fail"

print(f"Marks: 55 - {check_pass_fail(55)}")
print(f"Marks: 35 - {check_pass_fail(35)}")
print(f"Marks: 40 - {check_pass_fail(40)}")
print(f"Marks: 78 - {check_pass_fail(78)}")
```

```
Marks: 55 - Pass
Marks: 35 - Fail
Marks: 40 - Pass
Marks: 78 - Pass
```

Q2. Define a function min_max(numbers) that returns the minimum and maximum values from a list.

In [2]:

```
def min_max(numbers):
    minimum = min(numbers)
    maximum = max(numbers)
    return minimum, maximum

numbers = [12, 45, 7, 89, 23, 56, 3]
min_val, max_val = min_max(numbers)
print(f"List: {numbers}")
print(f"Minimum: {min_val}")
print(f"Maximum: {max_val}")
```

```
List: [12, 45, 7, 89, 23, 56, 3]
Minimum: 3
Maximum: 89
```

Q3. Create a class that acts as an iterator returning the Fibonacci series up to 100.

In [3]:

```
class Fibonacci:
    def __init__(self):
        self.a = 0
        self.b = 1

    def __iter__(self):
        return self

    def __next__(self):
        if self.a > 100:
            raise StopIteration
        current = self.a
        self.a, self.b = self.b, self.a + self.b
        return current

print("Fibonacci series up to 100:")
fib = Fibonacci()
for num in fib:
    print(num, end=" ")
```

```
Fibonacci series up to 100:
0 1 1 2 3 5 8 13 21 34 55 89
```

Q4. Use a lambda function with map() to square all elements in the list [1, 2, 3, 4, 5]

In [4]:

```
numbers = [1, 2, 3, 4, 5]
squared = list(map(lambda x: x ** 2, numbers))

print(f"Original list: {numbers}")
print(f"Squared list: {squared}")
```

```
Original list: [1, 2, 3, 4, 5]
Squared list: [1, 4, 9, 16, 25]
```

Q5. Use list comprehension to create a list of prime numbers between 1 and 50.

In [5]:

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

primes = [num for num in range(1, 51) if is_prime(num)]
print(f"Prime numbers between 1 and 50: {primes}")
```

```
Prime numbers between 1 and 50: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37,
41, 43, 47]
```

Q6. Write a function that accepts variable keyword arguments using **kwargs.

In [6]:

```
def display_info(**kwargs):
    print("Information received:")
    for key, value in kwargs.items():
        print(f"{key}: {value}")

display_info(name="Amit", age=21, city="Mumbai", course="Python")
```

```
Information received:
name: Amit
age: 21
city: Mumbai
course: Python
```