Amit Divekar | Assignment 2: Set A

Functions, Iterators and Generators - Set A

Q1. Write a function area_of_circle(radius) that calculates and returns the area of a circle (Use 3.14 as the value of π).

In [1]:

```python
def area_of_circle(radius):
    pi = 3.14
    area = pi * radius * radius
    return area

radius = 5
print(f"Area of circle with radius {radius} is: {area_of_circle(radius)}")

radius = 10
print(f"Area of circle with radius {radius} is: {area_of_circle(radius)}")
```

```
Area of circle with radius 5 is: 78.5
Area of circle with radius 10 is: 314.0
```

Q2. Create a lambda function to calculate the sum, difference, product and cube of numbers.

In [2]:

```python
add = lambda x, y: x + y
subtract = lambda x, y: x - y
multiply = lambda x, y: x * y
cube = lambda x: x ** 3

a = 10
b = 5

print(f"Sum of {a} and {b}: {add(a, b)}")
print(f"Difference of {a} and {b}: {subtract(a, b)}")
print(f"Product of {a} and {b}: {multiply(a, b)}")
print(f"Cube of {a}: {cube(a)}")
print(f"Cube of {b}: {cube(b)}")
```

```
Sum of 10 and 5: 15
Difference of 10 and 5: 5
Product of 10 and 5: 50
Cube of 10: 1000
Cube of 5: 125
```

Q3. Write a recursive function to calculate sum of digits of number till single digit. Example 457=16=7

In [3]:

```
def sum_of_digits(n):
    if n < 10:
        return n
    else:
        digit_sum = 0
        while n > 0:
            digit_sum += n % 10
            n = n // 10
        return sum_of_digits(digit_sum)

number = 457
print(f"Sum of digits of {number} till single digit: {sum_of_digits(number)}

number = 9875
print(f"Sum of digits of {number} till single digit: {sum_of_digits(number)}
```

```
Sum of digits of 457 till single digit: 7
Sum of digits of 9875 till single digit: 2
```

Q4. Create a function productInfo() that takes a product name and price and display the names of products whose price is less than Rs.100. If price is not given, assume the price is 100.

In [4]:

```
def productInfo(name, price=100):
    if price < 100:
        print(f"{name} - Rs.{price}")

productInfo("Pen", 50)
productInfo("Book", 150)
productInfo("Notebook", 80)
productInfo("Pencil", 10)
productInfo("Eraser")
```

```
Pen - Rs.50
Notebook - Rs.80
Pencil - Rs.10
```

Q5. Write a function that uses duck typing to process different data types.

In [5]:

```
def process_data(data):
    print(f"Length of data: {len(data)}")
    print(f"Data: {data}")
    print()

process_data("Hello")
process_data([1, 2, 3, 4, 5])
process_data((10, 20, 30))
process_data({"name": "Amit", "age": 21})
```

```
Length of data: 5
Data: Hello

Length of data: 5
Data: [1, 2, 3, 4, 5]

Length of data: 3
Data: (10, 20, 30)

Length of data: 2
Data: {'name': 'Amit', 'age': 21}
```

Q6. Write a generator that yields even numbers up to 10.

In [6]:

```
def even_numbers():
    for i in range(0, 11, 2):
        yield i

print("Even numbers up to 10:")
for num in even_numbers():
    print(num)
```

```
Even numbers up to 10:
0
2
4
6
8
10
```