**WEEK 4:**

**Exercise 1: Customer Directory Management System   
Objective   
The goal of this case study is to develop a Python program that manages customer information for a telecommunications company. This involves creating a function to handle customer data, extracting relevant information, and printing the data in various formats.   
  
Requirements   
1. Function Definition:   
A. Define a function named tel\_directory that takes a list of dictionaries representing customer information. Each dictionary contains:   
a. customer\_id (Unique identifier for the customer)   
b. customer\_name (Name of the customer)   
c. Subscription\_type (Type of subscription: "prepaid" or "postpaid")   
  
2. Data Input:   
B. Input at least 10 customer records using the tel\_directory function.   
  
3. Data Extraction:   
C. Extract data from the list of dictionaries and create a list of lists. Each list should be structured based on different key combinations.   
  
4. Output:   
D. Print the extracted data in three different combinations of key fields:   
a. Combination 1: A list containing [customer\_id, customer\_name, Subscription\_type] b. Combination 2: A list containing [customer\_id, customer\_name]   
c. Combination 3: A list containing [customer\_name, Subscription\_type]   
  
Exercise 2: Enhancing the Customer Directory Management System   
  
Add a new function named Search\_Customer to the existing Customer Directory Management System.   
This function should:   
1. Function Purpose:  
 A. Search for a customer by their name within the directory.   
2. Function Details:  
 A. The function should take the customer\_name as input.  
 B. If the customer is found in the directory, display their information.   
 C. If the customer is not found, print a message indicating that the customer is not in the directory.   
  
Exercise 3: Enhancing the Customer Directory Management System   
  
Add a new function named Search\_subscription to the existing Customer Directory Management System. This function should:  
  
1. Function Purpose:   
 A. Search for customers based on their subscription type.   
2. Function Details:   
 A. The function should take the Subscription\_type ("prepaid" or "postpaid") as input.   
 B. Display the information of all customers who have the specified subscription type.   
 C. If no customers have the given subscription type, print appropriate message.**

def tel\_directory(customers):

combo1 = [[cust['customer\_id'], cust['customer\_name'], cust['subscription\_type']] for cust in customers]

combo2 = [[cust['customer\_id'], cust['customer\_name']] for cust in customers]

combo3 = [[cust['customer\_name'], cust['subscription\_type']] for cust in customers]

print("Combination 1: [customer\_id, customer\_name, subscription\_type]")

print(combo1)

print("Combination 2: [customer\_id, customer\_name]")

print(combo2)

print("Combination 3: [customer\_name, subscription\_type]")

print(combo3)

def Search\_Customer(customers, customer\_name):

found = False

for cust in customers:

if cust['customer\_name'].lower() == customer\_name.lower():

print(f"Customer Found: ID: {cust['customer\_id']}, Name: {cust['customer\_name']}, Subscription Type: {cust['subscription\_type']}")

found = True

break

if not found:

print("Customer not found in the directory.")

def Search\_subscription(customers, subscription\_type):

found = False

for cust in customers:

if cust['subscription\_type'].lower() == subscription\_type.lower():

print(f"Customer Found: ID: {cust['customer\_id']}, Name: {cust['customer\_name']}, Subscription Type: {cust['subscription\_type']}")

found = True

if not found:

print("No customers found with the specified subscription type.")

def main():

customers = []

num\_customers = int(input("Enter the number of customers to add: "))

for \_ in range(num\_customers):

customer\_id = input("Enter customer ID: ")

customer\_name = input("Enter customer name: ")

subscription\_type = input("Enter subscription type: ")

# Adding customer information to the list

customers.append({

'customer\_id': customer\_id,

'customer\_name': customer\_name,

'subscription\_type': subscription\_type

})

# Display the different combinations

tel\_directory(customers)

# Search for a customer by name

search\_name = input("Enter customer name to search: ")

Search\_Customer(customers, search\_name)

# Search for customers by subscription type

search\_subscription = input("Enter subscription type to search: ")

Search\_subscription(customers, search\_subscription)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**OUTPUT:**

Enter the number of customers to add: 2

Enter customer ID: 101

Enter customer name: MAYANK

Enter subscription type: prepaid

Enter customer ID: 102

Enter customer name: Nishant

Enter subscription type: postpaid

Combination 1: [customer\_id, customer\_name, subscription\_type]

[['101', 'MAYANK', 'prepaid'], ['102', 'Nishant', 'postpaid']]

Combination 2: [customer\_id, customer\_name]

[['101', 'MAYANK'], ['102', 'Nishant']]

Combination 3: [customer\_name, subscription\_type]

[['MAYANK', 'prepaid'], ['Nishant', 'postpaid']]

Enter customer name to search: MAYANK

Customer Found: ID: 101, Name: MAYANK, Subscription Type: prepaid

Enter subscription type to search: prepaid

Customer Found: ID: 101, Name: MAYANK, Subscription Type: prepaid