

# **Week - 1 Deliverable Submission:-**

**Write a simple Python program that:**

1. Defines a Customer class

Fields:

id: int  
name: str  
email: str  
active: bool

Includes constructor, getters (or direct attributes), and `__repr__()`.

Solution:

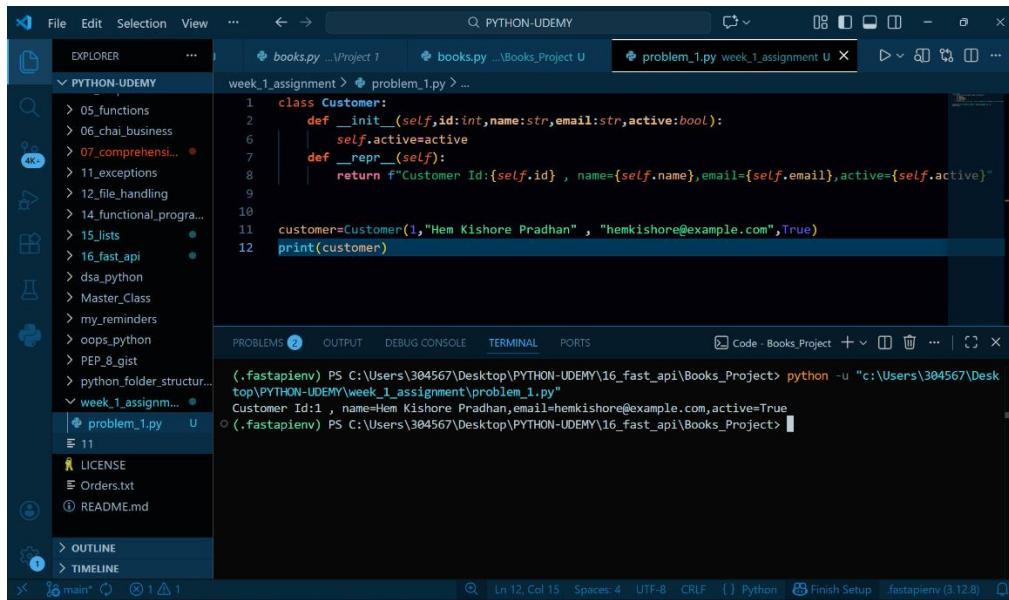
**class Customer:**

```
def __init__(self,id:int,name:str,email:str,active:bool):
    self.id=id
    self.name=name
    self.email=email
    self.active=active
def __repr__(self):
    return f"Customer Id:{self.id} ,
name={self.name},email={self.email},active={self.active}"
```

```
customer=Customer(1,"Hem Kishore Pradhan" , "hemkishore@example.com",True)
```

Output:

```
print(customer)
```



```
class Customer:
    def __init__(self,id:int,name:str,email:str,active:bool):
        self.id=id
        self.name=name
        self.email=email
        self.active=active
    def __repr__(self):
        return f"Customer Id:{self.id} , name={self.name},email={self.email},active={self.active}"
customer=Customer(1,"Hem Kishore Pradhan" , "hemkishore@example.com",True)
print(customer)
```

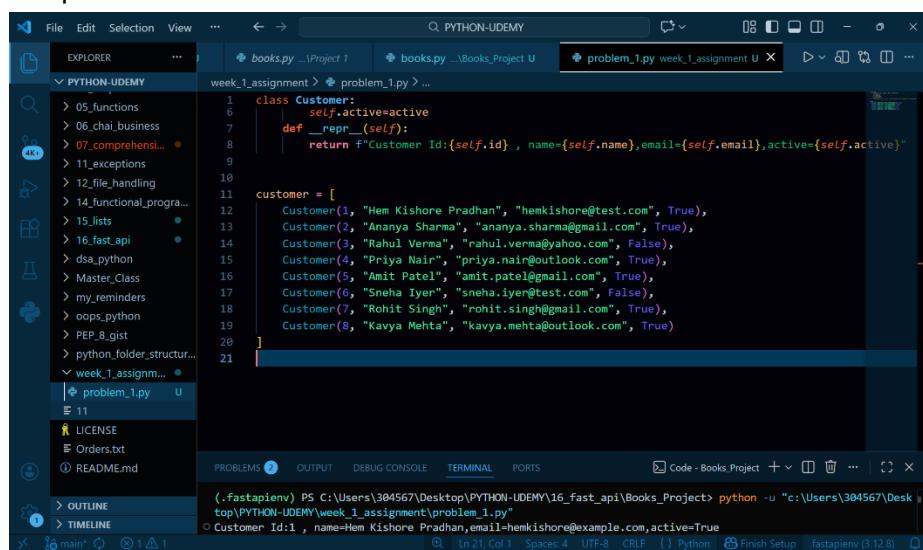
(.fastapienv) PS C:\Users\304567\Desktop\PYTHON-UDEMY\16\_fast\_api\Books\_Project> python -u "c:/Users/304567/Desktop/PYTHON-UDEMY/week\_1\_assignment/problem\_1.py"
Customer Id:1 , name=Hem Kishore Pradhan,email=hemkishore@example.com,active=True

## 2. Maintains a list of customers

solution:

```
customer = [
    Customer(1, "Hem Kishore Pradhan", "hemkishore@test.com", True),
    Customer(2, "Ananya Sharma", "ananya.sharma@gmail.com", True),
    Customer(3, "Rahul Verma", "rahul.verma@yahoo.com", False),
    Customer(4, "Priya Nair", "priya.nair@outlook.com", True),
    Customer(5, "Amit Patel", "amit.patel@gmail.com", True),
    Customer(6, "Sneha Iyer", "sneha.iyer@test.com", False),
    Customer(7, "Rohit Singh", "rohit.singh@gmail.com", True),
    Customer(8, "Kavya Mehta", "kavya.mehta@outlook.com", True)
]
```

Output:



```
class Customer:
    def __init__(self,id:int,name:str,email:str,active:bool):
        self.id=id
        self.name=name
        self.email=email
        self.active=active
    def __repr__(self):
        return f"Customer Id:{self.id} , name={self.name},email={self.email},active={self.active}"
customer = [
    Customer(1, "Hem Kishore Pradhan", "hemkishore@test.com", True),
    Customer(2, "Ananya Sharma", "ananya.sharma@gmail.com", True),
    Customer(3, "Rahul Verma", "rahul.verma@yahoo.com", False),
    Customer(4, "Priya Nair", "priya.nair@outlook.com", True),
    Customer(5, "Amit Patel", "amit.patel@gmail.com", True),
    Customer(6, "Sneha Iyer", "sneha.iyer@test.com", False),
    Customer(7, "Rohit Singh", "rohit.singh@gmail.com", True),
    Customer(8, "Kavya Mehta", "kavya.mehta@outlook.com", True)
]
```

(.fastapienv) PS C:\Users\304567\Desktop\PYTHON-UDEMY\16\_fast\_api\Books\_Project> python -u "c:/Users/304567/Desktop/PYTHON-UDEMY/week\_1\_assignment/problem\_1.py"
Customer Id:1 , name=Hem Kishore Pradhan,email=hemkishore@example.com,active=True

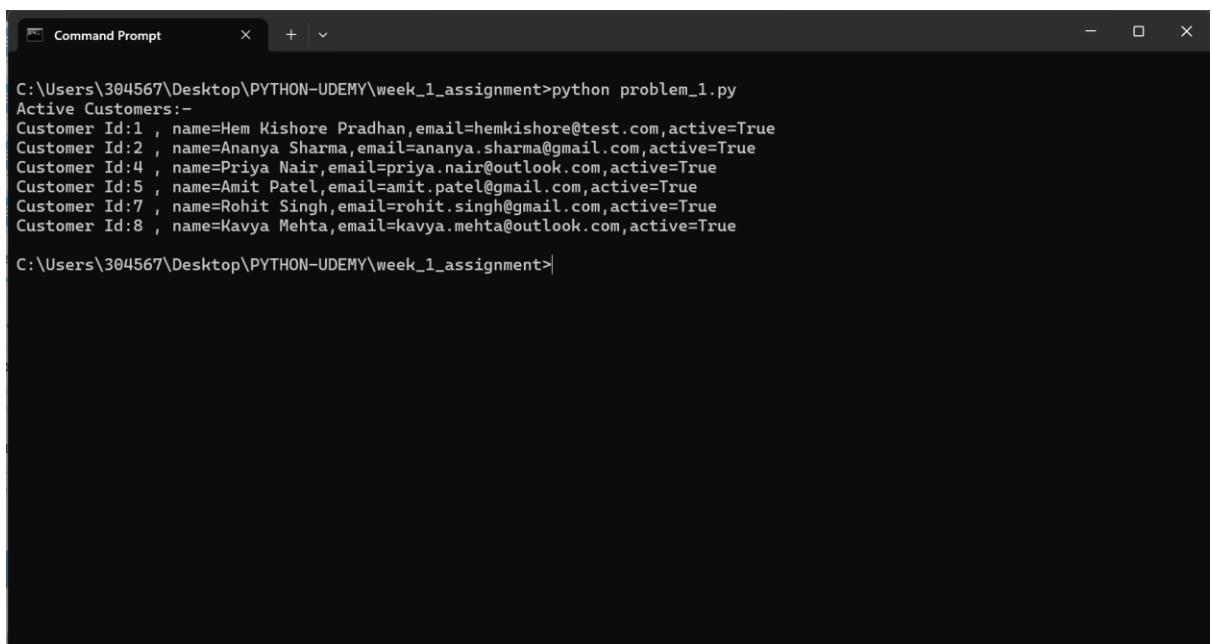
3. Implements data operations using Pythonic patterns

- a) Filter active customers using list comprehension

Solution:

```
active_customers=[c for c in customer if c.active]
```

output:



```
C:\Users\304567\Desktop\PYTHON-UDEMY\week_1_assignment>python problem_1.py
Active Customers:-
Customer Id:1 , name=Hem Kishore Pradhan,email=hemkishore@test.com,active=True
Customer Id:2 , name=Ananya Sharma,email=ananya.sharma@gmail.com,active=True
Customer Id:4 , name=Priya Nair,email=priya.nair@outlook.com,active=True
Customer Id:5 , name=Amit Patel,email=amit.patel@gmail.com,active=True
Customer Id:7 , name=Rohit Singh,email=rohit.singh@gmail.com,active=True
Customer Id:8 , name=Kavya Mehta,email=kavya.mehta@outlook.com,active=True

C:\Users\304567\Desktop\PYTHON-UDEMY\week_1_assignment>
```

- b) Search customers by name (case-insensitive)

solution:

```
search_customer_input=str(input("Enter the customer name : "))
for search_customer in customer:
    if search_customer.name.casefold() == search_customer_input.casefold():
        print(search_customer)
```

output:

```
C:\Users\304567\Desktop\PYTHON-UDEMY\week_1_assignment>python problem_1.py
Enter the customer name :
Customer Id:1 , name=Hem Kishore Pradhan,email=hemkishore@test.com,active=True
C:\Users\304567\Desktop\PYTHON-UDEMY\week_1_assignment>
```

c) Sort customers by name using sorted(key=...)

solution:

```
sorted_customer=sorted(customer,key= lambda c:c.name)
```

output:

```
C:\Users\304567\Desktop\PYTHON-UDEMY\week_1_assignment>python problem_1.py
[Customer Id:5 , name=Amit Patel,email=amit.patel@gmail.com,active=True, Customer Id:2 , name=Ananya Sharma,email=ananya.sharma@gmail.com,active=True, Customer Id:1 , name=Hem Kishore Pradhan,email=hemkishore@test.com,active=True, Customer Id:8 , name=Kavya Mehta,email=kavya.mehta@outlook.com,active=True, Customer Id:4 , name=Priya Nair,email=priya.nair@outlook.com,active=True, Customer Id:3 , name=Rahul Verma,email=rahul.verma@yahoo.com,active=False, Customer Id:7 , name=Rohit Singh,email=rohit.singh@gmail.com,active=True, Customer Id:6 , name=Sneha Iyer,email=sneha.iyer@test.com,active=False]
C:\Users\304567\Desktop\PYTHON-UDEMY\week_1_assignment>
```

d) Print results in a clean format

solution:

```
class Customer:
```

```
    def __init__(self, id: int, name: str, email: str, active: bool):
        self.id = id
        self.name = name
```

```

    self.email = email
    self.active = active

def __repr__(self):
    status = "Active" if self.active else "Inactive"
    return f"Customer Id: {self.id}, Name: {self.name}, Email: {self.email},
Status: {status}"

def filter_active_customers(customers):
    return [c for c in customers if c.active]

def search_customer_by_name(name, customers):
    for customer in customers:
        if customer.name.casfold() == name.casfold():
            return customer
    return None

def sort_customers_by_name(customers):
    return sorted(customers, key=lambda c: c.name)

def print_customers(customers):
    if not customers:
        print("No customers found.")
    for c in customers:
        print(c)

# Customer data
customers = [
    Customer(1, "Hem Kishore Pradhan", "hemkishore@test.com", True),
    Customer(2, "Ananya Sharma", "ananya.sharma@gmail.com", True),
    Customer(3, "Rahul Verma", "rahul.verma@yahoo.com", False),
    Customer(4, "Priya Nair", "priya.nair@outlook.com", True),
    Customer(5, "Amit Patel", "amit.patel@gmail.com", True),
    Customer(6, "Sneha Iyer", "sneha.iyer@test.com", False),
    Customer(7, "Rohit Singh", "rohit.singh@gmail.com", True),
    Customer(8, "Kavya Mehta", "kavya.mehta@outlook.com", True)
]

# Menu-driven program
while True:
    print("\n===== CUSTOMER MANAGEMENT MENU =====")

```

```

print("1. View all customers")
print("2. View active customers")
print("3. Search customer by name")
print("4. Sort customers by name")
print("5. Exit")

choice = input("Enter your choice (1-5): ")

if choice == "1":
    print("\n***** ALL CUSTOMERS *****")
    print_customers(customers)

elif choice == "2":
    print("\n***** ACTIVE CUSTOMERS *****")
    active_customers = filter_active_customers(customers)
    print_customers(active_customers)

elif choice == "3":
    name = input("Enter customer name to search: ")
    result = search_customer_by_name(name, customers)
    if result:
        print("\nCustomer Found:")
        print(result)
    else:
        print("\nCustomer not found.")

elif choice == "4":
    print("\n***** SORTED CUSTOMERS (BY NAME) *****")
    sorted_customers = sort_customers_by_name(customers)
    print_customers(sorted_customers)

elif choice == "5":
    print("Exiting program. Goodbye!")
    break

else:
    print("Invalid choice. Please select between 1 and 5.")

```

Output:

```
Command Prompt - python f + x
C:\Users\304567\Desktop\PYTHON-UDEMY\week_1_assignment>python problem_1.py

===== CUSTOMER MANAGEMENT MENU =====
1. View all customers
2. View active customers
3. Search customer by name
4. Sort customers by name
5. Exit
Enter your choice (1-5): 1

**** ALL CUSTOMERS ****
Customer Id: 1, Name: Hem Kishore Pradhan, Email: hemkishore@test.com, Status: Active
Customer Id: 2, Name: Ananya Sharma, Email: ananya.sharma@gmail.com, Status: Active
Customer Id: 3, Name: Rahul Verma, Email: rahul.verma@yahoo.com, Status: Inactive
Customer Id: 4, Name: Priya Nair, Email: priya.nair@outlook.com, Status: Active
Customer Id: 5, Name: Amit Patel, Email: amit.patel@gmail.com, Status: Active
Customer Id: 6, Name: Sneha Iyer, Email: sneha.iyer@test.com, Status: Inactive
Customer Id: 7, Name: Rohit Singh, Email: rohit.singh@gmail.com, Status: Active
Customer Id: 8, Name: Kavya Mehta, Email: kavya.mehta@outlook.com, Status: Active

===== CUSTOMER MANAGEMENT MENU =====
1. View all customers
2. View active customers
3. Search customer by name
4. Sort customers by name
5. Exit
Enter your choice (1-5): 4

**** SORTED CUSTOMERS (BY NAME) ****
Customer Id: 5, Name: Amit Patel, Email: amit.patel@gmail.com, Status: Active
```

#### 4. Write a pseudo-REST contract in comments

Solution:

*# Pseudo REST API Contract*

*# GET /customers*

*# Description: Fetch all customers*

*# GET /customers?active=True*

*# Description: Fetch only active customers*

*# GET /customers/search?name=abc*

*# Description: Search a customer by name.*