

OOPS DAY-2

1. Question 1 – JSONPlaceholder Users → Class Objects

Create a class User that stores data fetched from

<https://jsonplaceholder.typicode.com/users>.

Requirements:

1. The constructor (__init__) should accept id, name, and email.
2. Define a method showUser() to print the user details in a readable format.
 - o Example: User #1 → Leanne Graham (Sincere@april.biz)
3. Define another method getEmailDomain() to return only the domain part of the email.
 - o Example: april.biz
4. Fetch users from the API and create multiple User objects.
5. Print details of at least 5 users using showUser() and getEmailDomain().

Code:-

```
import requests
apiUrl="https://jsonplaceholder.typicode.com/users"
response=requests.get(apiUrl)

class User:
    def __init__(self,id,name,email):
        self.id=id
        self.name=name
        self.email=email

    def showUser(self):
        print(f"User #{self.id} -> {self.name} ({self.email})")

    def getEmailDomain(self):
        return self.email.split('@')[-1]

if response.status_code==200:
    data=response.json()
    user_objects=[User(u["id"], u["name"], u["email"]) for u in data]

    for i in user_objects[:5]:
        i.showUser()
        print("Domain:",i.getEmailDomain())
else:
    print("Failed to fetch data from API")
```

Output:-

User #1 -> Leanne Graham (Sincere@april.biz)
Domain: april.biz
User #2 -> Ervin Howell (Shanna@melissa.tv)
Domain: melissa.tv
User #3 -> Clementine Bauch (Nathan@yesenia.net)
Domain: yesenia.net
User #4 -> Patricia Lebsack (Julianne.OConner@kory.org)
Domain: kory.org
User #5 -> Chelsey Dietrich (Lucio_Hettinger@annie.ca)
Domain: [annie.ca](https://jsonplaceholder.typicode.com/)

2. Question 2 – JSONPlaceholder Posts → Blog System

Create a class Post to manage blog posts fetched from

<https://jsonplaceholder.typicode.com/posts>

Requirements:

1. The constructor should accept userId, id, title, and body.
2. Define a method showSummary() that prints the post ID and first 20 characters of the title.
o Example: Post #1 → sunt aut facere repellat...
3. Define another method getWordCount() that counts how many words are in the body.
4. Fetch posts from the API and create multiple Post objects.
5. For the first 3 posts, show summary and word count.

Code:-

```
import requests
apiUrl="https://jsonplaceholder.typicode.com/posts"
response=requests.get(apiUrl)

class Post:
    def __init__(self,ui,i,t,b):
        self.userid=ui
        self.id=i
        self.title=t
        self.body=b

    def showSummary(self):
        print(f"Post #{self.id} → {self.title[:20]}...")

    def getWordCount(self):
        return len(self.body.split())

if response.status_code==200:
    data=response.json()
    post_objects = [Post(p["userId"], p["id"], p["title"], p["body"]) for p in data]

    for post in post_objects[:3]:
        post.showSummary()
        print("Word Count:", post.getWordCount())
else:
```

```
print("Failed to fetch data from API")
```

Output:-

Post #1 → sunt aut facere repe...

Word Count: 23

Post #2 → qui est esse...

Word Count: 31

Post #3 → ea molestias quasi e...

Word Count: 26

3. Question 3 – DummyJSON Products → Flipkart Style

Use API: <https://dummyjson.com/products>

Create a class Product to store details of each product.

Requirements:

1. The constructor should accept id, title, price, and stock.
2. Define a method showDetails() to display product info. o Example: Laptop (₹50000) – Stock: 12
3. Define another method buyProduct(qty) that reduces stock if enough stock is available, otherwise print "Out of stock".
4. Fetch product data from API and create multiple Product objects.
5. Simulate a shopping cart by buying:
 - o 2 units of the first product.
 - o 1 unit of the second product.
 - o Try buying more than stock for the third product.

Code:-

```
import requests
```

```
class Product:
```

```
    def __init__(self, id, title, price, stock):
```

```
        self.id = id
```

```
        self.title = title
```

```
        self.price = price
```

```
        self.stock = stock
```

```
    def showDetails(self):
```

```
        print(f"{self.title} (₹{self.price}) – Stock: {self.stock}")
```

```
    def buyProduct(self, qty):
```

```
        if qty <= self.stock:
```

```
            self.stock -= qty
```

```
            total_cost = qty * self.price
```

```
            print(f"Bought {qty} x {self.title} → Total: ₹{total_cost}")
```

```
        else:
```

```
            print(f"{self.title}: Out of stock (Requested {qty}, Available {self.stock})")
```

```
url = "https://dummyjson.com/products"
```

```
response = requests.get(url)
```

```
if response.status_code == 200:
    products_data = response.json()["products"]
    product_objects = [Product(p["id"], p["title"], p["price"], p["stock"]) for p in products_data]
    print("Available Products:")
    for prod in product_objects:
        prod.showDetails()

    product_objects[0].buyProduct(2)
    product_objects[1].buyProduct(1)
    product_objects[2].buyProduct(9999)

else:
    print("Failed to fetch products")
```

Output:-

Available Products:

Essence Mascara Lash Princess (₹9.99) – Stock: 99
Eyeshadow Palette with Mirror (₹19.99) – Stock: 34
Powder Canister (₹14.99) – Stock: 89
Red Lipstick (₹12.99) – Stock: 91
Red Nail Polish (₹8.99) – Stock: 79
Calvin Klein CK One (₹49.99) – Stock: 29 Chanel
Coco Noir Eau De (₹129.99) – Stock: 58 Dior
J'adore (₹89.99) – Stock: 98
Dolce Shine Eau de (₹69.99) – Stock: 4 Gucci
Bloom Eau de (₹79.99) – Stock: 91
Annibale Colombo Bed (₹1899.99) – Stock: 88
Annibale Colombo Sofa (₹2499.99) – Stock: 60
Bedside Table African Cherry (₹299.99) – Stock: 64
Knoll Saarinen Executive Conference Chair (₹499.99) – Stock: 26 Wooden
Bathroom Sink With Mirror (₹799.99) – Stock: 7
Apple (₹1.99) – Stock: 8
Beef Steak (₹12.99) – Stock: 86
Cat Food (₹8.99) – Stock: 46
Chicken Meat (₹9.99) – Stock: 97
Cooking Oil (₹4.99) – Stock: 10
Cucumber (₹1.49) – Stock: 84 Dog
Food (₹10.99) – Stock: 71 Eggs
(₹2.99) – Stock: 9
Fish Steak (₹14.99) – Stock: 74 Green
Bell Pepper (₹1.29) – Stock: 33 Green
Chili Pepper (₹0.99) – Stock: 3 Honey
Jar (₹6.99) – Stock: 34
Ice Cream (₹5.49) – Stock: 27
Juice (₹3.99) – Stock: 50
Kiwi (₹2.49) – Stock: 99
Bought 2 x Essence Mascara Lash Princess → Total: ₹19.98

Bought 1 x Eyeshadow Palette with Mirror → Total: ₹19.99

Powder Canister: Out of stock (Requested 9999, Available 89)