Key Concepts: Instance Variables and Methods in Python

Problem: Automating Payment Status for Employees

- Restaurant managers find it cumbersome to manually update each other about employee payment statuses.
- An automated system is needed to track and update payments for employees efficiently.

Solution: Using Instance Variables and Methods

Step 1: Create the Class

- Define a class called `PaySlips` .
- Initialize three instance variables: `name` , `payment` , and `amount` in the `__init__` method.

```
class PaySlips:
    def __init__(self, name, payment, amount):
        self.name = name
        self.payment = payment
        self.amount = amount
```

Step 2: Add Methods

- **`pay()` Method**:
  - Updates the `payment` status to `"yes"` .
  - Example:

    ```
    def pay(self):
        self.payment = "yes"
    ```

- **`status()` Method**:
  - Checks the `payment` status and returns an appropriate message.
  - Example:

    ```
    def status(self):
        if self.payment == "yes":
            return f"{self.name} is paid {str(self.amount)}"
        else:
            return f"{self.name} is not paid yet"
    ```

Step 3: Create Instances and Test the Code

Define Instances

- Create instances for employees, `Nathan` and `Roger` .

```
nathan = PaySlips("Nathan", "no", 1000)
roger = PaySlips("Roger", "no", 3000)
```

Check Payment Status

- Use the `status()` method to display the current payment status for each employee.

```
print(nathan.status())
print(roger.status())
```

Update Payment Status

- Use the `pay()` method to update Nathan's payment status and recheck the status.

```
nathan.pay()
print("\nAfter Payment:")
print(nathan.status())
print(roger.status())
```

Key Learnings

- **Instance Variables**:
  - Store unique data for each object (e.g., `name` , `payment` , `amount`).
  - Changes to one object's instance variables do not affect others.
- **Instance Methods**:
  - Operate on the data (state) of the specific object they are called on.
  - Allow encapsulated logic that modifies or retrieves object data.

Advantages of This System

- Each employee's payment status can be updated independently without affecting others.
- Simplifies the process for managers by automating repetitive tasks.
- Lays the foundation for an online payment system where managers can click a "Paid" button to update the status.