Classes and Objects

Class

A **class** is a blueprint or template used to create objects.
It defines **properties (variables)** and **behaviors (methods)**.
Example:
A BankAccount class defines what an account *has* (balance, name) and *can do* (deposit, withdraw).

Object

An **object** is a real instance of a class.
Each object has its **own data**.
Example:
account1 = BankAccount("Athar", 5000)

Simple Example

```
class Student:
    def __init__(self, name, marks):
        self.name = name
        self.marks = marks

    def display(self):
        print("Name:", self.name)
        print("Marks:", self.marks)

# Object creation
s1 = Student("Athar", 85)
s1.display()
```

Basic Concepts of Inheritance and Methods

Method

A **method** is a function defined inside a class that performs an action.

```
def deposit(self, amount):
    self.balance += amount
```

Inheritance

Inheritance allows one class to acquire properties and methods of another class.

- **Parent class (Base class)**
- **Child class (Derived class)**

Helps in **code reuse**

Inheritance Example

```
class Person:
    def __init__(self, name):
        self.name = name
    def show_name(self):
        print("Name:", self.name)

class Student(Person):
    def __init__(self, name, roll_no):
        super().__init__(name)
        self.roll_no = roll_no

    def show_details(self):
        print("Roll No:", self.roll_no)

s = Student("Athar", 101)
s.show_name()
s.show_details()
```

Activity: Build a Simple Class-Based Project
Bank Account Simulator (Recommended)

This project uses:

- Class
- Object
- Methods
- Inheritance

Complete Bank Account Project (Python)

```
class BankAccount:
    def __init__(self, holder_name, balance=0):
        self.holder_name = holder_name
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount
        print(f"Deposited {amount}. New balance: {self.balance}")

    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
            print(f"Withdrawn {amount}. Remaining balance: {self.balance}")
        else:
            print("Insufficient balance")

    def show_balance(self):
        print("Account Holder:", self.holder_name)
        print("Balance:", self.balance)


# Inheritance example
class SavingsAccount(BankAccount):
    def __init__(self, holder_name, balance, interest_rate):
        super().__init__(holder_name, balance)
        self.interest_rate = interest_rate

    def add_interest(self):
        interest = self.balance * self.interest_rate / 100
        self.balance += interest
        print("Interest added:", interest)


# Object creation
account1 = SavingsAccount("Athar Ahmed", 5000, 5)

account1.show_balance()
account1.deposit(2000)
account1.withdraw(1000)
account1.add_interest()
account1.show_balance()
```