

Concept: Object-Oriented Programming (OOP)

OOP is a programming paradigm that organizes code into classes and objects. Classes define the blueprint, while objects are instances of those classes.

Core principles: encapsulation, inheritance, and polymorphism. Encapsulation hides internal details; inheritance allows class reuse; polymorphism enables methods with the same name to behave differently.

Classes use the `def __init__()` constructor for initializing object attributes.

Using OOP makes code modular, maintainable, and reusable.

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

    def display(self):
        print(f'Name: {self.name}, Age: {self.age}')

s1 = Student('Ava', 20)
s1.display()

class Graduate(Student):
    def __init__(self, name, age, degree):
        super().__init__(name, age)
        self.degree = degree

    def show(self):
        print(f'{self.name} graduated in {self.degree}')
```