ADVANCED PROGRAMMING II- Object-Oriented Programming (OOP) Basics

Introduction to Object-Oriented Programming

Object-Oriented Programming (OOP) is a programming approach that organises a program around **objects** rather than just functions and procedures.

An **object** represents something that has:

- **Properties (attributes)** — what the object has or its characteristics.
- **Methods** — what the object can do or its actions.

Example

Think about a **Student**.

A student may have:

Properties:

- Name
- Age
- Class
- Admission number

Methods:

- Register()
- Study()
- TakeExam()
- DisplayResult()

In OOP, these properties and methods can be grouped together into an object.

What is an Object?

An **object** is an instance of a **class**.

For example:

A class can be thought of as a blueprint, while an object is something created from that blueprint.

If **Student** is a class, then:

- David can be an object.
- Mary can be an object.
- John can be an object.

They are all students, but each object can have different values.

Object	Name	Age	Class
Student 1	David	15	SS1
Student 2	Mary	14	SS1
Student 3	John	16	SS1

What is a Class?

A **class** is a blueprint or template used to create objects.

It defines the properties and behaviours that objects of that type will have.

Example

A `Student` class might contain:

```
Class Student

Properties:
    name
    age
    class

Methods:
    study()
    takeExam()
    displayResult()
```

The class describes what a student object should contain and what it can do.

Class and Object

The difference can be remembered using a simple example.

Class = Blueprint

Imagine an architect creates a blueprint for a house.

Object = Actual House

Several houses can be built from that blueprint.

Similarly:

Class → Blueprint

Object → Instance created from the blueprint

Properties / Attributes

Properties, also called **attributes**, are characteristics or pieces of information belonging to an object.

For a `Car` object, properties might include:

- Brand
- Colour
- Model
- Speed
- Year

For example:

```
Car
    brand = Toyota
    colour = Blue
    model = Corolla
    year = 2025
```

Methods

A **method** is a function that belongs to a class or object.

Methods describe what an object can **do**.

For example, a `Car` may have:

```
start()
stop()
accelerate()
brake()
```

A `Student` may have:

```
study()
writeExam()
read()
```

Basic OOP Example

Here is a simple example using Python:

```
class Student:    def __init__(self, name, age):        self.name = name        self.age = age    def introduce(self):        print("My name is", self.name)student1 = Student("David", 15)student1.introduce()
```

Explanation

`class Student:`

Creates a class called `Student`.

`def __init__(self, name, age):`

This is a special method used when creating a new object.

`self.name = nameself.age = age`

These store the student's information.

`student1 = Student("David", 15)`

Creates an object called `student1`.

`student1.introduce()`

Calls the `introduce()` method.