Tops Technology

## Module 15) Advance Python Programming

Presented By:

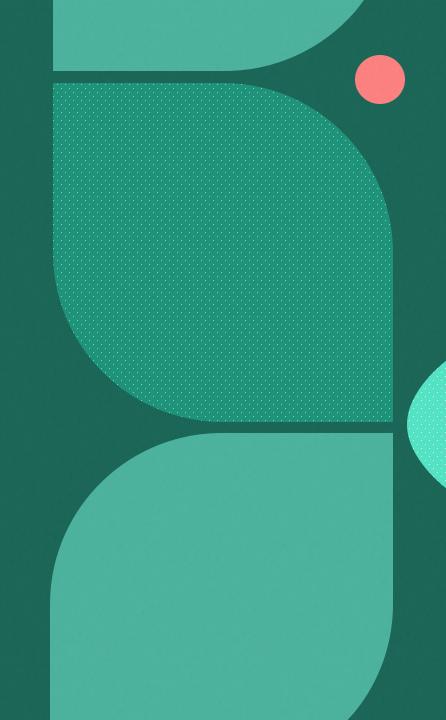
Nandni Vala

## Class and Object (OOP Concepts)

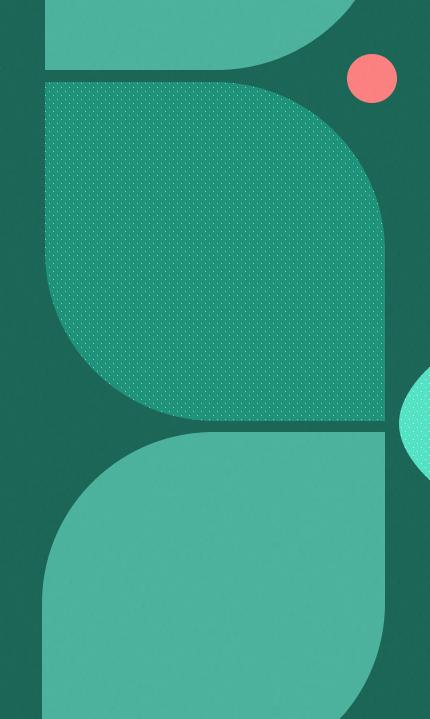
- ➤ 1.Understanding the concepts of classes, objects, attributes, and methods in Python.
- > Classes
- A class is a blueprint or template for creating objects.
- ➤ Defines the attributes (data) and methods (functions) that its objects will have.
- Example:
- Class Car:
- > # Class definition
- > pass
- > Objects
- > An **object** is an instance of a class.
- > Represents a specific realization of the class blueprint.
- > Example:
- > my\_car = Car() # Object of the Car class



- > Attributes
- ➤ Attributes are variables associated with a class or its objects.
- ➤ They represent the properties of an object (e.g., color, model, speed).
- > Attributes can be:
- ➤ **Instance Attributes**: Unique to each object.
- ➤ Class Attributes: Shared among all objects of the class.
- > Example:
- > class Car:
- $\triangleright$  wheels = 4 # Class attribute (common to all cars)
- def \_\_init\_\_(self, color, model):
- > self.color = color # Instance attribute
- > self.model = model # Instance attribute
- > # Creating objects
- $\triangleright$  car1 = Car("Red", "Sedan")
- > car2 = Car("Blue", "SUV")
- print(car1.color) # Output: Red (unique to car1)
- > print(Car.wheels) # Output: 4 (common for all cars)



- > Methods
- ➤ **Methods** are functions defined inside a class that operate on the attributes of the object.
- ➤ Common types of methods:
- ➤ **Instance Methods**: Operate on instance attributes.
- **class Methods**: Operate on class attributes.
- > Static Methods: General utility functions that do not operate on class or instance attributes.
- Example:
- > class Car:
- def \_\_init\_\_(self, color, model):
- $\triangleright$  self.color = color
- $\triangleright$  self.model = model
- def display\_info(self): # Instance method
- print(f''Color: {self.color}, Model: {self.model}'')
- > # Creating an object and calling a method
- $\triangleright$  car1 = Car("Red", "Sedan")
- car1.display\_info() # Output: Color: Red, Model: Sedan



## 2. Difference between local and global variables.

Feature	Local Variable	Global Variable
Defined in	Inside a function or block	Outside all functions or at the top level
Scope	Accessible only within the defining function	Accessible throughout the program
Lifetime	Created and destroyed within a function call	Exists for the entire program execution
Default Behavior	Local unless specified otherwise	Global unless shadowed by a local variable