1. What is a Class in Python and How is it Used to Create Objects?

A class in Python is a blueprint for creating objects. It defines attributes (variables) and methods (functions) that describe the behavior and properties of objects.

Create a Class and Use It to Create Objects?

```
class Car:
    def __init__(self, brand, model, price):
        self.brand = brand  # Attribute
        self.model = model
        self.price = price

    def display(self):
        print(f"Car: {self.brand} {self.model}, Price: {self.price}")

car1 = Car("Toyota", "Camry", 30000)
car1.display()
```

2. What are Methods and Attributes in Python Classes?

Attributes (Instance Variables)

Attributes are variables associated with an object.

They define the state or properties of an object.

Methods:

Methods define behavior of the object.

They are functions inside a class that perform actions.

3. What is Abstraction in OOP and How Does It Simplify Complex Systems?

Definition of Abstraction:

Abstraction is a principle of Object-Oriented Programming (OOP) that hides unnecessary implementation details and only shows essential features.

It simplifies complex systems by exposing only relevant functionalities while keeping the internal workings hidden.

How Does Abstraction Simplify Complex Systems?

Hides Complexity:

Users interact with only the required functionalities without worrying about implementation details.

Improves Code Maintainability:

Changes in internal logic don’t affect the users of the class.

Enhances Security:

Internal data is protected from direct access.