1. What is class in Python, and how it is used to create objects?

A. A **class** in Python is a blueprint or template used to create objects. It defines a set of attributes (variables) and methods (functions) that describe the behavior and properties of the objects created from it. Classes enable **Object-Oriented Programming (OOP)**, allowing developers to model real-world entities such as students, cars, or bank accounts.

To create an object (also called an **instance**) from a class, we use the class name followed by parentheses. Python automatically runs the `__init__()` method (constructor) to initialize object attributes. For example:

```
class Car:
    def __init__(self, brand, color):
        self.brand = brand
        self.color = color

mycar = Car("Toyota", "Red")
```

Here, `mycar` is an object with its own `brand` and `color` values. Classes help organize code better, promote reusability, and support modular design.

2. What are methods and attributes in Python Classes?

A. In Python classes, **attributes** are variables that store data related to an object. They represent the state or properties of an object. For example, a `Student` class may have attributes like `name`, `age`, and `roll_number`.

Methods are functions defined inside a class that describe the behaviors or actions an object can perform. Methods always take `self` as the first parameter, which refers to the specific object calling the method. Example:

```
class Student:
    def study(self):
        print("Student is studying")
```

Attributes store **data**, while methods define **behavior**. Together, they form the structure of an object in OOP.