

a. Difference between a Class and an Object in Python:

- **Class:**
 - A class is a blueprint or template for creating objects. It defines the properties (attributes) and behaviors (methods) that the objects created from it will have.
 - Think of a class as a blueprint for a house. You can use that blueprint to build multiple houses, each with the same structure but possibly different features.
 - In Python, a class is defined using the `class` keyword.
- **Object:**
 - An object is an instance of a class. When a class is defined, no memory is allocated until an object (or instance) of that class is created.
 - Objects have actual values for the properties and can use the behaviors defined in the class.
 - Continuing with the house analogy, each house built using the blueprint is an object, and while they share the same structure, their features (like paint color or flooring) may differ.

b. Example:

```
python
Copy code
# Defining a class
class Dog:
    def __init__(self, name, breed):
        self.name = name # Attribute
        self.breed = breed # Attribute

    def bark(self):
        print(f"{self.name} is barking!") # Method (behavior)

# Creating objects (instances) of the class
dog1 = Dog("Buddy", "Golden Retriever") # Object 1
dog2 = Dog("Max", "Bulldog") # Object 2

# Accessing methods and attributes of objects
```

```
print(dog1.name) # Output: Buddy  
print(dog2.breed) # Output: Bulldog  
  
dog1.bark() # Output: Buddy is barking!  
dog2.bark() # Output: Max is barking!
```