1. **CLASS**

- A class in Java is a set of objects which shares common characteristics/ behavior and common properties/ attributes.
- It is a user-defined blueprint or prototype from which objects are created.
- Example, Student is a class while a particular student named Ravi is an object.
- Class is not a real-world entity. It is just a template or blueprint or prototype from which objects are created
- Class does not occupy memory.
- We can also say class is a factory which produces object for us.

Syntax to define class

Example .

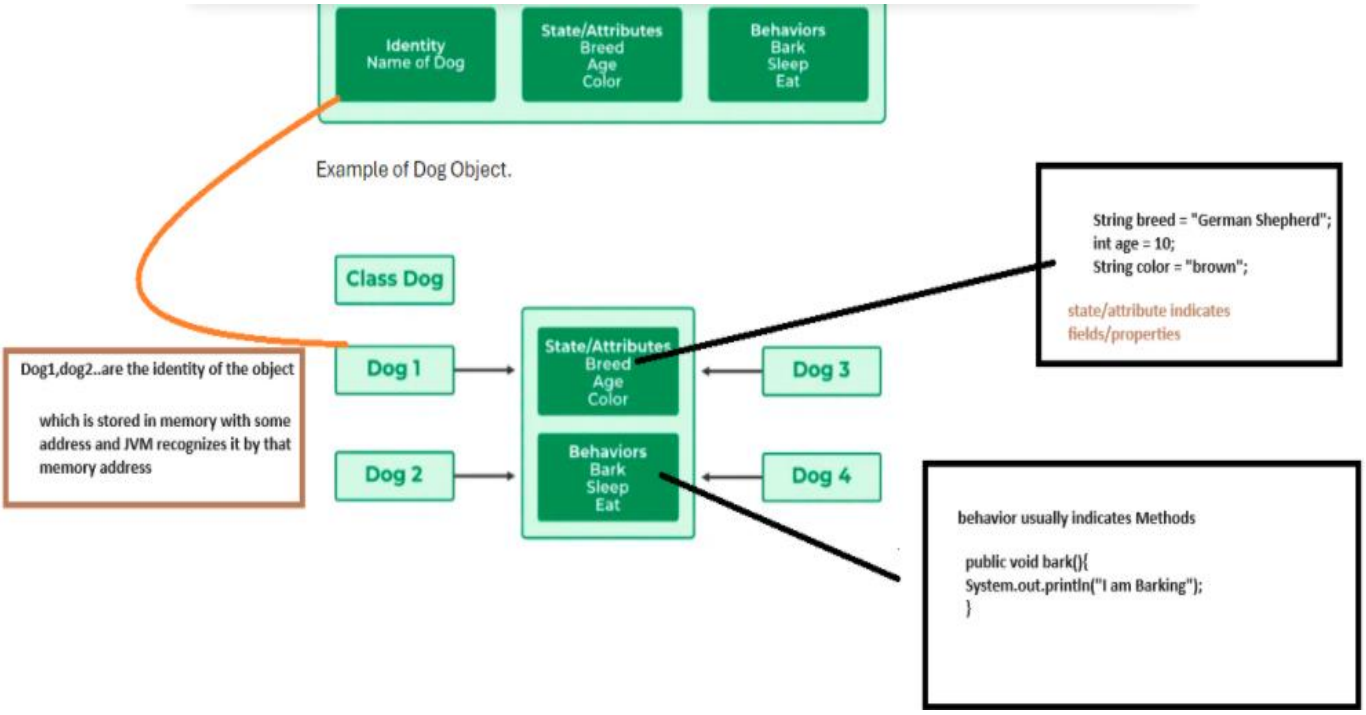
```java
class Student {
    // data member (also instance variable)
    int id;
    // data member (also instance variable)
    String name;

    public static void main(String args[])
    {
        // creating an object of
        // Student
        Student s1 = new Student();
        System.out.println(s1.id);
        System.out.println(s1.name);
    }
}
```

---

2. Objects

- An object in Java is a basic unit of Object-Oriented Programming and represents real-life entities.
- Objects are the instances of a class.
- We can create any number of object
- Objects are created in heap memory.
- Object variables are Non- Primitive Data type .
- An object Consists of
    - **State**: It is represented by attributes of an object. It also reflects the properties of an object.
    - **Behavior**: It is represented by the methods of an object. It also reflects the response of an object with other objects.
    - **Identity**: It gives a unique name to an object and enables one object to interact with other objects

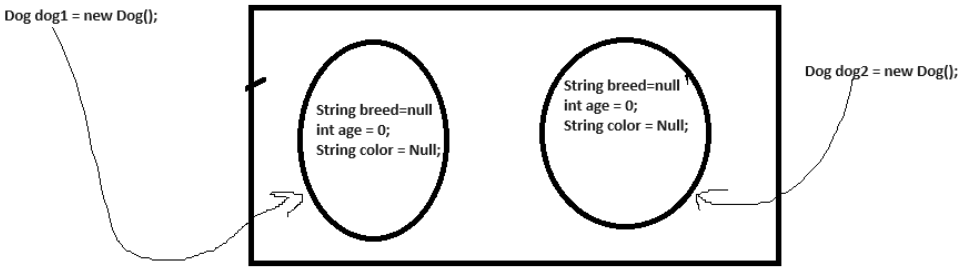


Synatx to Create Object

here className() is called constructor.

ClassName variableName = new ClassName();

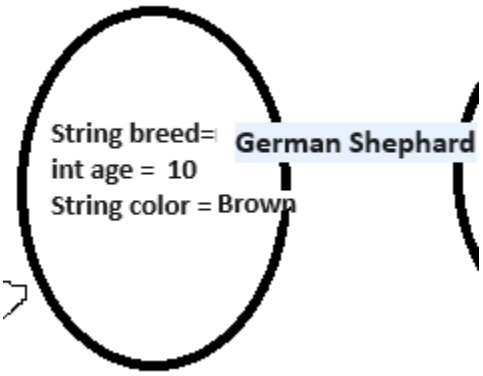
With the help of new keyword ,the request goes to class to create a new Object.



Whenever we create an object all the non static variables gets initialized for all the objects with default value if variables are not initialized.

```java
public class Dog {
    String breed;
    int age;
    String color;
    //all the above variables are called as state of object
    public static void main(String[] args) {
        Dog dog1 = new Dog();
        // here when dog1 we create , the attributes breed
//      age and color will be initialized with its default
//      values
    System.out.println(dog1.age);//print 0 (int default value
    System.out.println(dog1.breed);//print null;
//      Now we assign new value to the dog1
        dog1.age=10;
        dog1.breed= "German Shephard";
        dog1.color= "Brown";
        System.out.println(dog1.age);//prints 10
    System.out.println(dog1.breed);//prints German Shephard
```

Now this is how dog1 will be updated



Example 2

```java
public class Dog {
    String breed = "Indian Breed";
    int age = 6;
    String color = "Black";
    //all the above variables are called as state of object
    public static void main(String[] args) {
        Dog dog1 = new Dog();
        // here when dog1 we create , the attributes breed
//      age and color will be initialized
//      with the values we given while declaring var
    System.out.println(dog1.age);//print 6
    System.out.println(dog1.breed);//print "Indian Breed";
//      Now we can reassign value to the dog1
        dog1.age=10;
        dog1.breed= "German Shephard";
        dog1.color= "Brown";
        System.out.println(dog1.age);//prints 10
    System.out.println(dog1.breed);//prints German Shephard
```

In this case the objects will be initialized with value provided initially , not with default value