Core Java

Unit II – Constructor Overloading, Static Variables & Methods

Observe the code given below and answer the questions

```
class Sample
{
    private static void display(int a)
{
 System.out.println("Arguments: " + a);
    }
    private static void display(int a, int b)
{
    System.out.println("Arguments: " + a + " and " + b);
    }
    public static void main(String[] args)
{
        display(1);
        display(1, 4);
    }   }
```

Which concept of OOPs is implemented in the below code?

- Polymorphism (Method Overloading)

What type of Polymorphism is implemented in the given code?

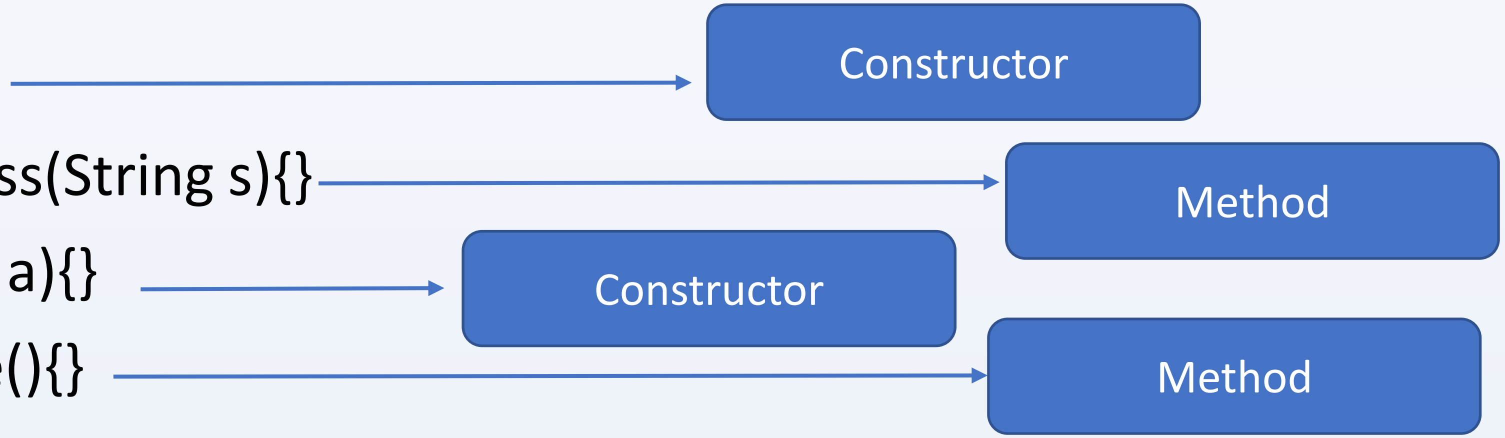
- Compile Time Polymorphism

Why it is known as Compile-Time Polymorphism?

- Since all the method definitions should be known at the time of compilation.

Review

```
class MyClass
{
        public MyClass(){}
        public void  MyClass(String s){}
        public MyClass(int a){}
        public int calculate(){}
}
```



Constructor

Method

Constructor

Method

Learning Outcomes

- Constructor Overloading
- this keyword
- Static Variables
- Static Methods

Constructor Overloading

Constructor overloading in Java is a technique of having more than one constructor with different parameter lists.

They are arranged in a way that each constructor performs a different task.

Example

class Account{

- Account(int a);
- Account (int a,int b);
- Account (String a,int b);

}

this keyword

- **“this”** keyword can be used inside any method to refer to the **current** object.
- That is, **this** is always a reference to the object on which the method was invoked.
- When a local variable has the same name as an instance variable, the local variable hides the instance variable.
- So, to refer to the instance variable this keyword is used.
- Example:

```
class box
{
 double width,height,depth;
 Box(double width, double height, double depth) {
 this.width = width;
 this.height = height;
 this.depth = depth;
} }
```

Static Variable

The static variable can be used to refer the common property of all objects (that is not unique for each object) e.g. company name of employees, college name of students etc.

The static variable gets memory only once in class area at the time of class loading.

It is also known as Class Variable.

Example :- static int a=20;

Static Method

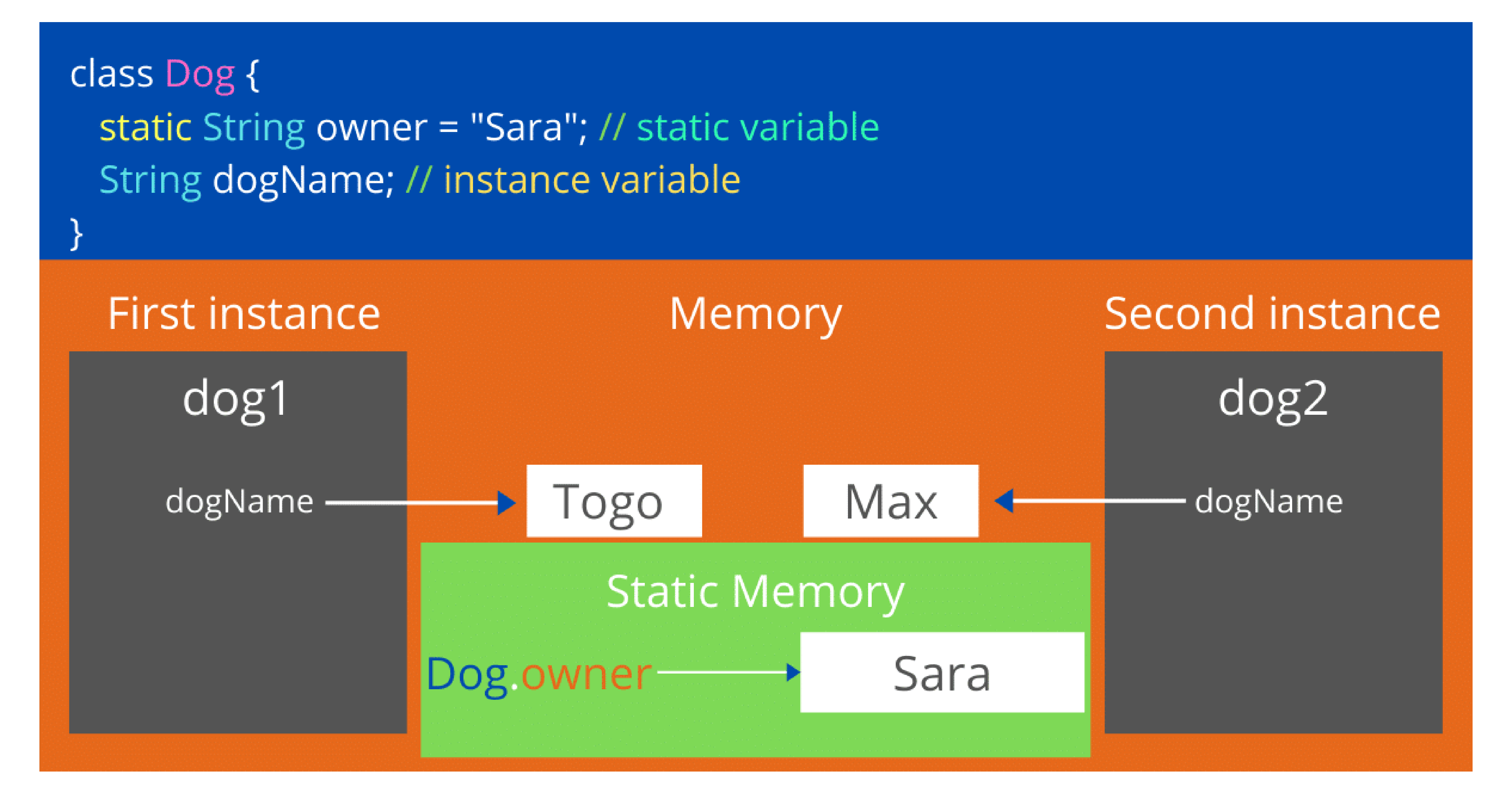
A static method belongs to the class rather than object of a class.

A static method can be invoked without the need for creating an instance of a class.

Static method can access only static data member and can change the value of it.

Example:-

- static void getData(){} // method definition
- getData() // calling of method



```
class Dog {
   static String owner = "Sara"; // static variable
   String dogName; // instance variable
}
```

First instance

Memory

Second instance

dog1

dogName

Togo

Max

dog2

dogName

Static Memory

Dog.owner

Sara

Quiz

Thank You