

The background is a vibrant, abstract composition. It features large, organic shapes in teal, pink, yellow, and black. These shapes are decorated with various patterns: white dots, white wavy lines, white plus signs, and black wavy lines. The overall style is modern and graphic.

CORE JAVA

UNIT II – METHODS & CONSTRUCTORS

LEARNING OUTCOMES

METHOD OVERLOADING

VARARGS

CONSTRUCTORS & ITS TYPES

CONSTRUCTOR OVERLOADING

METHOD OVERLOADING

If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.



Concept of Polymorphism.



There are two ways to overload the method in java

By changing number of arguments

By changing the data type

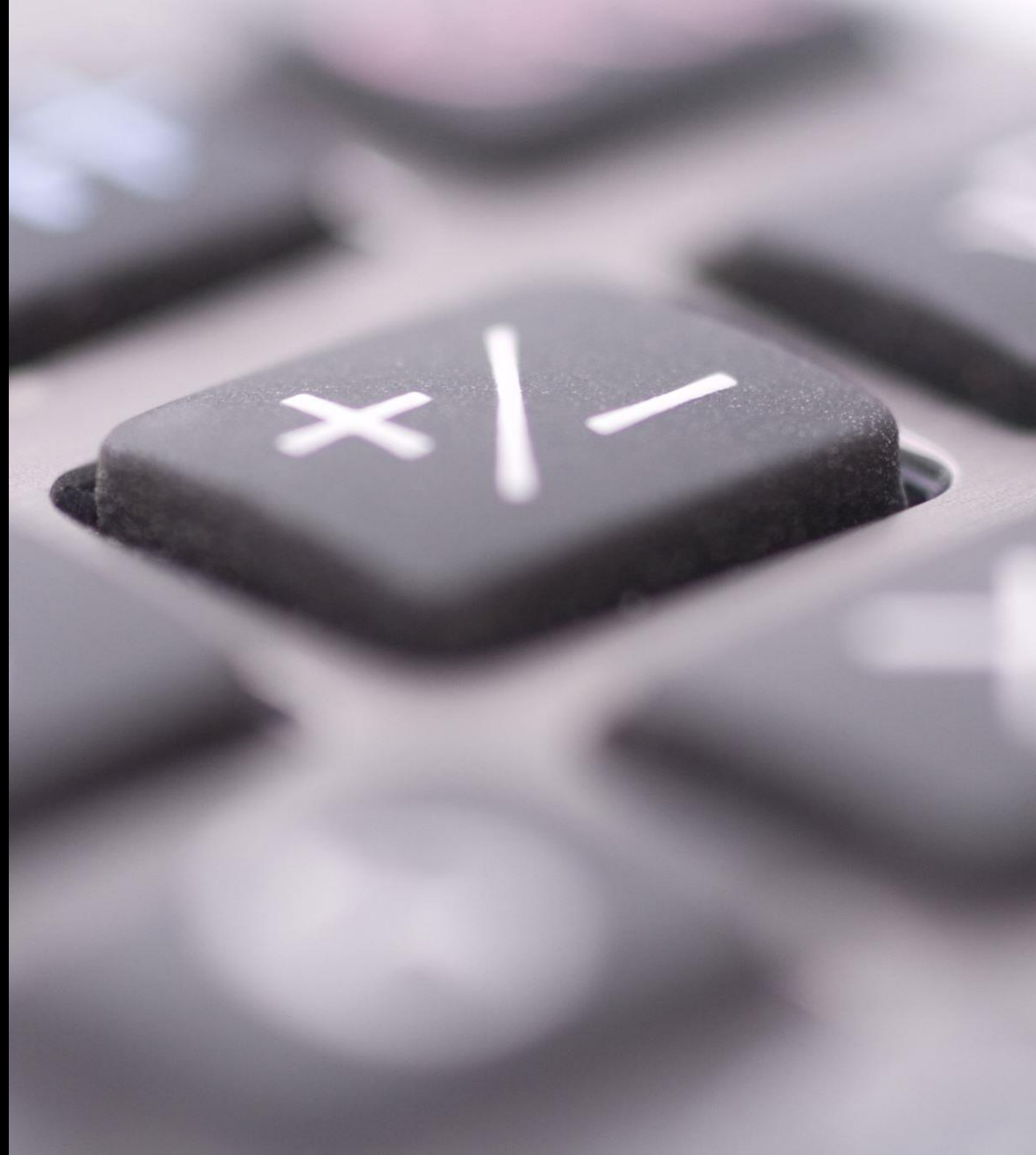
Changing number of Arguments

```
class Adder{  
    int add(int a,int b){  
        return a+b;  
    }  
    int add(int a,int b,int c)  
    {  
        return a+b+c;  
    }  
}
```



Changing the Data Type

```
class Adder{  
    int add(int a, int b)  
    {  
        return a+b;  
    }  
    double add(double a, double b)  
    {  
        return a+b;  
    }  
}
```



VARIABLE ARGUMENTS

Allows declaring a method that can accept a variable number of parameters for a given argument.

Vararg must be the last argument in the formal argument list.

SYNTAX:

```
return_type method_name(data_type... variableName)
{
    //body of method
}
```

EXAMPLE:

```
public void show(String... records){ } // definition  
    show("VSIT","VP","VIT"); // Method Calling
```

CONSTRUCTORS

01

Special type of a method that enables an object to initialize itself when it is created.

02

Same name as the class itself.

03

Do not specify return type, not even void.

04

EXAMPLE:

```
class Rectangle
{
    Rectangle(){ }    //
    constructor
}
```


TYPES OF CONSTRUCTORS

DEFAULT

- When it doesn't have any parameter.

PARAMETERIZED

- Having a specific number of parameters is called a parameterized constructor.

COPY


- creates an object using another object of the same Java class.

CONSTRUCTOR OVERLOADING

```
class Box
{
    double w,h,d;
    Box()
    {
        w = -1;
        h = -1;
        d = -1;
    }
    Box(double l)
    {
        w=h=d=l;
    }
}
```

QUIZ





Thank You.