**5. Methods in Java**

• **Theory:**

1. **Defining Methods: -** In Java, methods are blocks of code that perform a specific task and can be reused throughout your program. Here's a breakdown of how to define methods in Java, including their syntax.

**Syntax:-**

accessModifier returnType methodName(parameterType parameterName) {

// method body

// return statement (if returnType is not void)

}

**2.Method Parameters and Return Types:-**

**Method Parameters:-**

Parameters are variables listed in a method's definition that allow the method to accept input values when called.

**Types of Parameters:-**

1. Required Parameters: Must be provided when calling the method.
2. Optional Parameters: Can be given default values (Java doesn't support optional parameters directly, but you can use method overloading to achieve similar functionality).
3. Variable Arguments (Varargs): Allows you to pass a variable number of arguments to a method.

**Syntax:-** public void methodName(type1 param1, type2 param2) {

// Method body }

**3.Method Overloading:-** Method overloading in Java allows you to define multiple methods with the same name but different parameter lists within the same class.

**Rules for Method Overloading:-**

1. Different Parameter List: The methods must have different parameter lists.
2. Same Method Name: The method name must remain the same.
3. Return Type: The return type can be the same or different, but it alone cannot be used to differentiate overloaded methods.

**Syntax:-** returnType methodName(parameterType1 parameterName1, parameterType2 parameterName2, ...) { // method body // return statement (if applicable) }

**4.Static Methods and Variables:-** A static method belongs to the class rather than any particular object. It can be called without creating an instance of the class. Static methods can only access static variables and other static methods directly.

**Syntax:-** class ClassName {

static returnType methodName(parameters) {

// method body

}

}