The line public static void main(String args[]) { } is the entry point for the execution of a Java program.

It is a special method that serves as the starting point for the program's execution when it is run.

Here's a breakdown of each component in the line:

- public: It is an access modifier that allows the main method to be accessed from outside the class. It indicates that the method can be called by any other class.
- static: It is a keyword that indicates that the main method belongs to the class itself and not to any instance of the class. This allows the method to be called without creating an object of the class.
- void: It is the return type of the main method, which means it does not return any value.
- main: It is the name of the method. The Java Virtual Machine (JVM) looks for this specific method as the entry point to start the program's execution.
- (String args[]): It is the parameter list of the main method. It accepts a single parameter, an array of String objects, which can be used to pass command-line arguments to the Java program.

Inside the main method, you write the code that defines the program's behavior. This is where you write statements and instructions that will be executed when the program runs.

For example, a simple Java program with the main method might look like this:

Java

```
public class HelloWorld {
    public static void main(String args[]) {
        System.out.println("Hello, World!");
    }
}
```

In this example, when the program is executed, it will print “Hello, World!” to the console because of the System.out.println statement inside the main method.

The main method is crucial in Java as it provides the starting point for the program’s execution and allows you to run and test your Java code.

Java in Hindi Video

Related Posts:

1. Can Java have same name variable
2. Types of variables in Java programming
3. JAVA and its Support Systems
4. JAVA environment
5. JAVA program structure
6. Tokens
7. Java statements
8. Java virtual machine
9. C++ Versus JAVA
10. Constants and Variables in Java
11. Data types JAVA

12. Defining a class
13. Constructor in JAVA
14. Array in Java
15. Applet
16. Applets Vs Applications
17. Writing applets
18. Applets life cycle
19. Creating an Executable Applet
20. Graphics in Applet
21. Applet image display
22. Applet digital clock
23. Applet mouse event handling
24. JDBC
25. Execute an SQL Statement
26. Process the result
27. CLOSE THE DATABASE CONNECTION
28. File handling
29. Define a class to declare an integer array of size n and accept the elements into the array.
30. Define a class to declare an array of size 20 of the double datatype, accept the elements into the array and perform the following: Calculate and print the sum of all the elements.
31. Java program for String, to uppercase, to equal, length of string
32. Write a Java program for Buble sort.
33. Write a Java program String to uppercase and count words startig with ‘A’
34. How to set path in Java
35. Difference between static and non static methods in Java