Day 04 – Command-Line Arguments, JDK, JRE & JVM

1. Command-Line Arguments

Command-line arguments are values passed to a Java program at the time of execution. They are received through the String[] args parameter of the main() method.

Example:

```
class Demo {
public static void main(String[] args) {
System.out.println(args[0]);
}
}
```

```
Execution:
java Demo Hello
Output: Hello
```

Diagram:

Command Line → JVM → main(String[] args) → Program Execution

2. JDK (Java Development Kit)

JDK is used to develop Java applications. It includes the compiler (javac), development tools, and JRE.

JDK Diagram:

```
JDK
■
■■■ JRE
■ ■■■ JVM
■ ■■■ Libraries
■■■ Development Tools
```

3. JRE (Java Runtime Environment)

JRE provides the environment to run Java programs. It contains JVM and core libraries but does not include development tools.

JRE Diagram:

```
JRE
■
■■■ JVM
■■■ Libraries
```

4. JVM (Java Virtual Machine)

JVM executes Java bytecode and converts it into machine-specific code. It also manages memory and security.

JVM Working Diagram:

Java Source (.java)
↓
Compiler (javac)
↓
Bytecode (.class)
↓
JVM
↓
Machine Code

Difference between JDK, JRE and JVM

Component	Purpose
JDK	Develop Java applications
JRE	Run Java applications
JVM	Execute bytecode

Conclusion

Command-line arguments and JDK, JRE, JVM are core concepts that explain how Java programs are created and executed.