The previous content has been cleared, and the new **“Introduction to Java”** note has been placed into the Canvas:

Introduction to Java

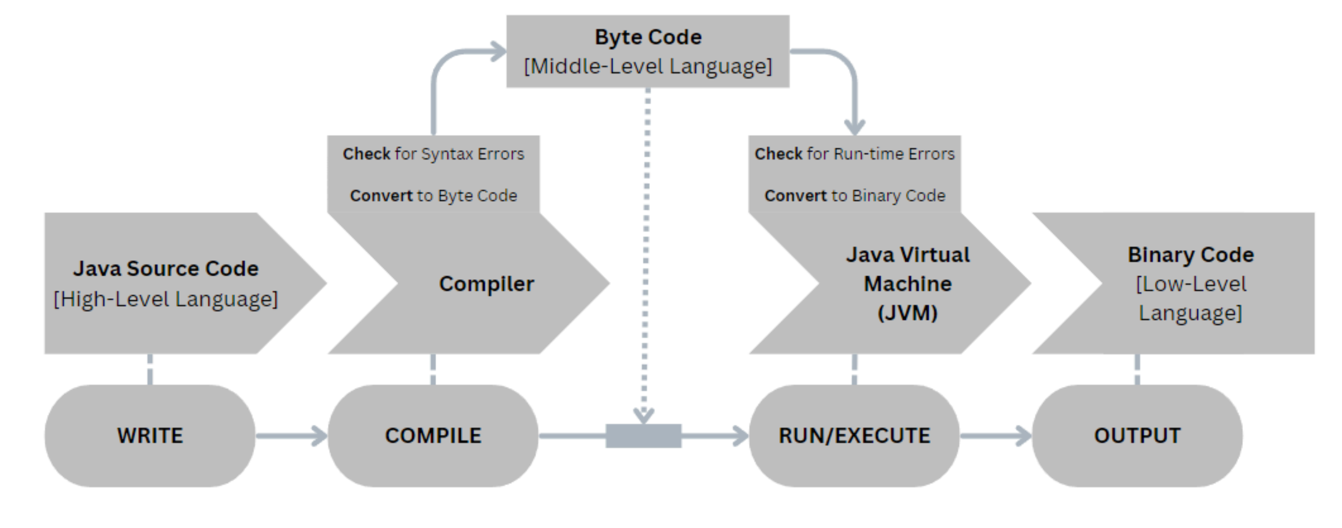
What is Java?

Java is a *programming language* that enables the **development** and **testing** of software applications.

---

1. What is Happening?

Java Source Code Execution Procedure



The code written by the programmer is called the **Java Source Code (JSC)**. This code is not directly understood by the system. To make it executable, two core software components are involved:

1. **Compiler**
2. **Java Virtual Machine (JVM)**

Compiler

- A **compiler** is a software tool that reads and processes the Java Source Code.
- Its responsibilities include:
    - **Checking** the syntax (format) of the code.

    - **Converting** the JSC into **Byte Code (ByC)**.
    - **Forwarding** the Byte Code to the JVM.

> *Note: The JVM can only interpret and execute Byte Code.*

Java Virtual Machine (JVM)

- The **JVM** is a runtime environment that executes the Byte Code.
- It performs the following actions:
    - **Receives** the Byte Code.
    - **Checks** for runtime errors.
    - **Converts** the Byte Code into **Binary Code (BiC)**, also known as **Machine Code**.

> *Note: The processor understands only Binary Code. Once the BiC is received, the system processes it and produces the output.*

To carry out Java operations efficiently, a software environment known as an **IDE** is typically used.

---

2. Where is it Happening?

Integrated Development Environment (IDE)

An **IDE (Integrated Development Environment)** provides a platform where developers can write, compile, and execute Java programs.

Examples of IDEs:

- **Basic Code Editors:**
    1. Notepad
    2. Notepad++
    3. EditPlus
- **Advanced IDEs:**
    1. Eclipse
    2. Visual Studio Code
    3. IntelliJ IDEA

4. Sublime Text

> *Note: To execute Java programs within any IDE, the **Java Development Kit (JDK)** must be installed.*

---

3. Who Will Be of Help?

Java Development Kit (JDK)

- The **JDK** provides a collection of tools and libraries required to perform Java-related operations.
- It enables the compilation and execution of Java code within any IDE or terminal.

---

4. Using What Does the Process Work?

Major Components of Java:

1. **Class** – blueprint for objects
2. **Method** – block of code that performs a task
3. **Variable** – container to store data
4. **Object** – instance of a class

Basic Syntax Example:

```java
public class MessagePrinter {
    public static void main(String[] args) {
        System.out.println("Good Night !! Doston");
    }
}
```

Explanation:

- `public class MessagePrinter { ... }` → **Class declaration**
- `public static void main(String[] args) { ... }` → **Main method**
- `System.out.println("...");` → **Print statement**

---

5. What Needs to Be Done?

Java Development Process:

WRITE → SAVE → COMPILE → RUN → OUTPUT

---

6. How is it Done?

To generate output from Java code, follow these steps:

1. **Write**
   - Enter the source code in a text editor or IDE.
2. **Save**
   - Save the file with a `.java` extension, e.g., `MessagePrinter.java`.
3. **Compile**
   - Use the `javac` command to compile the file:

     ```
     javac path\to\MessagePrinter.java
     ```

   - After successful compilation, a `MessagePrinter.class` file (Byte Code) is created.
4. **Run**
   - Use the `java` command to execute the class:

     ```
     java path\to\MessagePrinter
     ```

   - The JVM converts the Byte Code to Binary Code and the system prints the output.
