**Difference Between Interpreter and JIT Compiler**

Both **Interpreter** and **JIT (Just-In-Time) Compiler** are part of the **JVM Execution Engine**, and their job is to **convert Java bytecode into machine code** so that the CPU can execute it. However, they work differently.

**1. Interpreter (Slow but starts execution quickly)**

**How it works?**

* Reads and executes **bytecode line by line**.
* **Translates the same code repeatedly** every time it is executed.
* Slower because it doesn’t save compiled machine code.

**Example:**  
Think of an interpreter as a **real-time translator** at a conference. Every time someone speaks, the translator **translates word by word** for the audience. This takes time.

**2. JIT (Just-In-Time) Compiler (Fast but takes time to optimize)**

**How it works?**

* Analyses code and **compiles frequently used methods into native machine code**.
* The compiled machine code is **saved and reused**, making execution much faster.
* Helps improve performance over time.

**Example:**  
Think of the JIT compiler as **preparing subtitles** for a movie. Instead of translating in real-time, subtitles are prepared **once** and can be **reused** every time someone watches the movie.