**Runtime Memory Areas in JVM**

The **Java Virtual Machine (JVM)** divides memory into different runtime areas to manage the execution of Java programs efficiently. The main memory areas are:

**1. Method Area (Metaspace in Java 8+)**

**Stores:**

* Class metadata (including fully qualified class names, method and field names, and bytecode).
* Runtime constant pool (including literals and references).
* Static variables.

**2. Heap Area**

**Stores:**

* **All objects and instance variables.**
* **It’s Shared** among all threads.
* Divided into **Young Generation (Eden + Survivor Spaces) and Old Generation (Tenured Space).**
* **Garbage Collection** manages unused objects.

**3. Stack Area**

**Stores:**

* **Method call frames** (local variables, parameters, return addresses).
* **Each thread gets its own stack** (not shared).
* **Causes StackOverflowError if it exceeds allocated memory.**

**4. PC Register (Program Counter Register)**

**Stores:**

* The **address of the currently executing JVM instruction**.
* **Each thread has its own PC register.**
* Helps in **switching between threads**.

**5. Native Method Stack (C Stack)**

**Stores:**

* Native method calls (methods written in C/C++ using JNI - Java Native Interface).
* Each thread has a separate **native stack**.
* Used for interfacing Java with **native libraries** (e.g., C/C++).