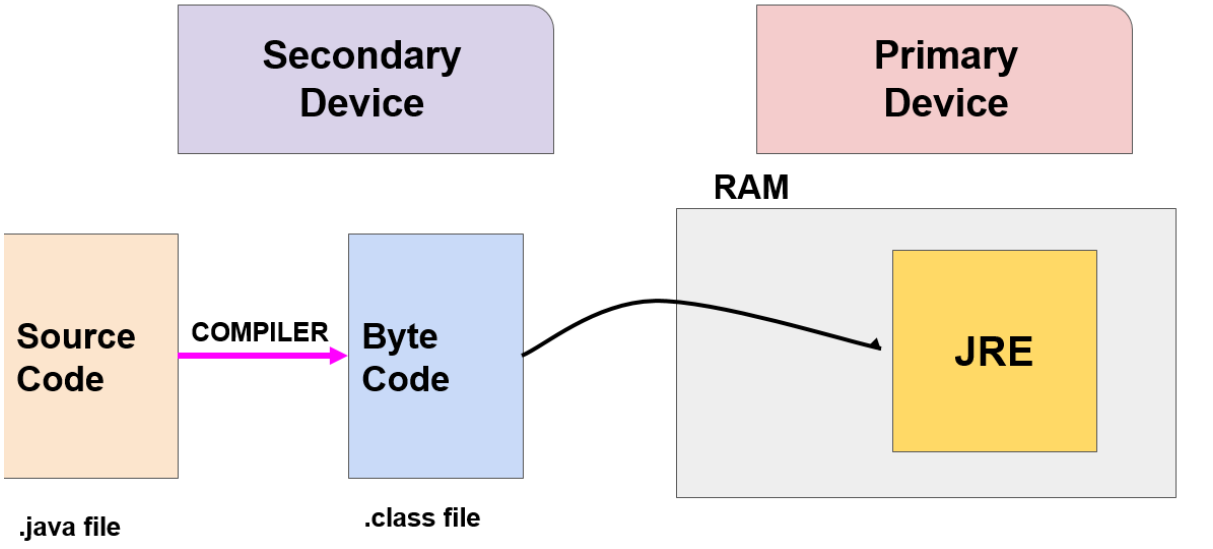
JAVA RUNTIME MEMORY ALLOCATION

- In Java we need a place where we can store our variables, methods and initializers.
- In java we also need a place to execute our program

Hence, Java provide us with Java Runtime Environment (JRE) to store and execute our program.

- JRE needs class file to perform its task.
- When we have to execute our program JRE will request main memory (RAM) to provide some memory for class loading process as well as program execution process.

JRE AND CLASS LOADING PROCESS



- To execute the java program a portion of memory in RAM is allocated for JRE.
- In that portion of memory allocated, we have different range of memory, hence they are classified as follows,

1. **Method area**
2. **Class static area**

3. **Stack area**
4. **Heap area**

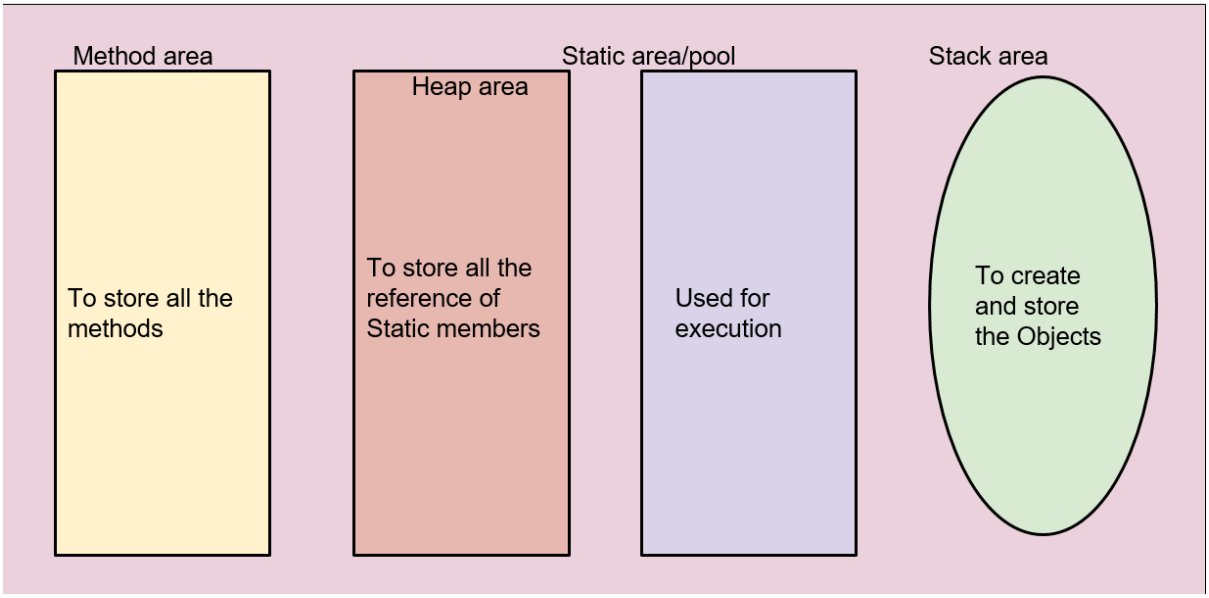
Loading a class file from secondary memory to JRE is called <u>Class loading Process.</u>

<u>CLASS LOADING PROCESS:</u>

- A block is created for a class in the static pool, it can be accessed with the help of a class name.
- All the method definitions are loaded in the method area and if the method is static then the reference of that method is stored inside the class block (class static area).
- If the class has any static variables they are loaded in the class static area with a default value.
- If the class has any static initializers they are executed from top to bottom order,
- The loading process of the class is completed, then JVM will call the main method of the initial loading class.

NOTE :

JVM will call only the main method of the initial loaded class.



METHOD AREA :

All the method blocks will be stored in a method area (Instruction of the methods).

CLASS STATIC AREA :

● For every class there is a dedicated block of memory is created in the class static area (static pool).
● The static members of the class will be allocated inside the memory created for the class.

STACK AREA :

● The stack area is used for the execution of instructions.

● For every method that is under execution a block of memory is created in this stack area which is known as a frame.

● Once the execution of a method is completed the frame is removed.

HEAP AREA :

● In a heap area, a block of memory is created for the instance of a class (Object).

● Every block of memory is created with the help of reference.

● All the non-static members of a class will be allocated inside this block of memory.

● Therefore, we can access the non-static member with the help of reference.