Linking :

Verify :

It **ensures the correctness of the .class files**, If any suspicious activity is there in the .class file then it will **stop the execution immediately** by throwing a runtime error i.e java.lang.VerifyError.

There is something called **ByteCodeVerifier**(Component of JVM), responsible to verify the loaded .class file i.e byte code. Due to this verify module JAVA is highly secure language.

java.lang.VerifyError is the sub class of java.lang.linkageError.

Prepare [static member memory allocation + initialized with default value**] :**

It will allocate the memory for all the Static Fields, here all the Static Fields will get the **default values** so if we have static int x = 100; then for variable x, memory will be allocated (4 bytes) and now it will initialize with default value i.e. 0, even the variable is final.

static Test t = new Test();

Here, t is a static reference field so for t field (reference variable) memory will be allocated as per JVM implementation i.e. for 32 bit JVM (4 bytes of Memory) and for 64 bit (8 bytes of memory) and initialized with null.

Resolve :

All the symbolic references (#7) will be converted into **direct references OR actual reference.**

javap -verbose FileName.class

Note :- By using above command we can read the internal details of .class file.

Initialization :

Here class initialization will takes place. All the static field member will get their actual/original value and we can also use static block for static field member initialization.

Here, In this class initialization phase static field and static block is having same priority so it will executed according to the order.(Top to bottom)

**Can we write a Java Program without main method ?

```
class WithoutMain
{
   static
   {
       IO.println("Hello User!!");
       System.exit(0);
   }
}
```

It was possible to write a java program without main method till **JDK 1.6V.**
From JDK 1.7v onwards, at the time of loading the .class file JVM will verify the **presence of main method** in the .class file. If main method is not available then it will generate a runtime error that "main method not found in so so class".

What is static block OR Static Initializer :
----
* It is a special block in java which is **automatically executed at the time of loading the .class file into JVM memory.**
  Example :
```
  static
  {
      //static block
  }
```

* Unlike non static block, which will be **executed everytime based on object creation, static block will be executed only one time because in Java class loading is possible only one time.**

* A static block will not be executed everytime, It depends whether .class file is loaded OR not [that means A static block will only execute when the .class file will be loaded]

  * There are 5 ways to load the .class file into JVM memory :

    a) By using java command
    b) By using Constructor (Object creation)
    c) By accessing any static member (static variable OR static Method) of the class
    d) By using Inheritance (Before sub class, **Super class will be loaded first**)
    e) By using Reflection API

* The main purpose of **static block to initialize the static data member** that is the reason It is also known as **Static Initializer.**

* If a class contains **multiple static blocks then It will be executed top to bottom [Order wise]**