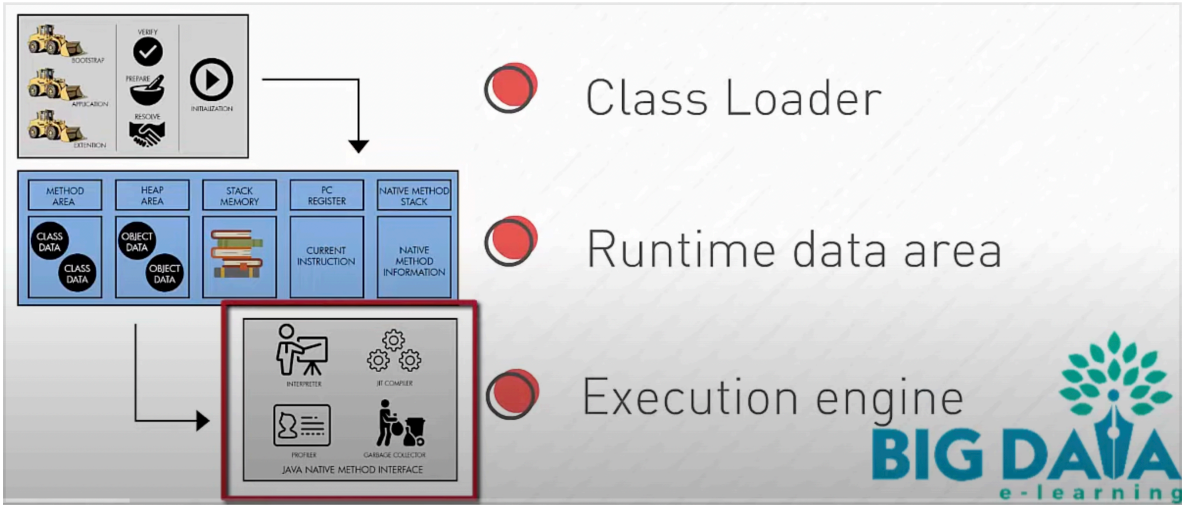
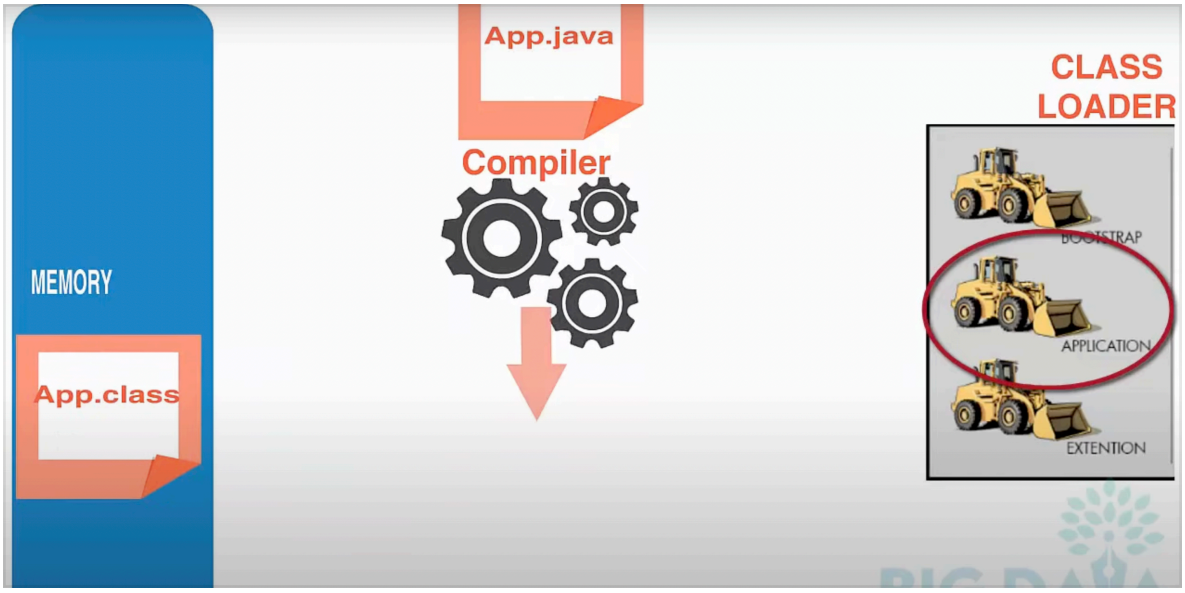
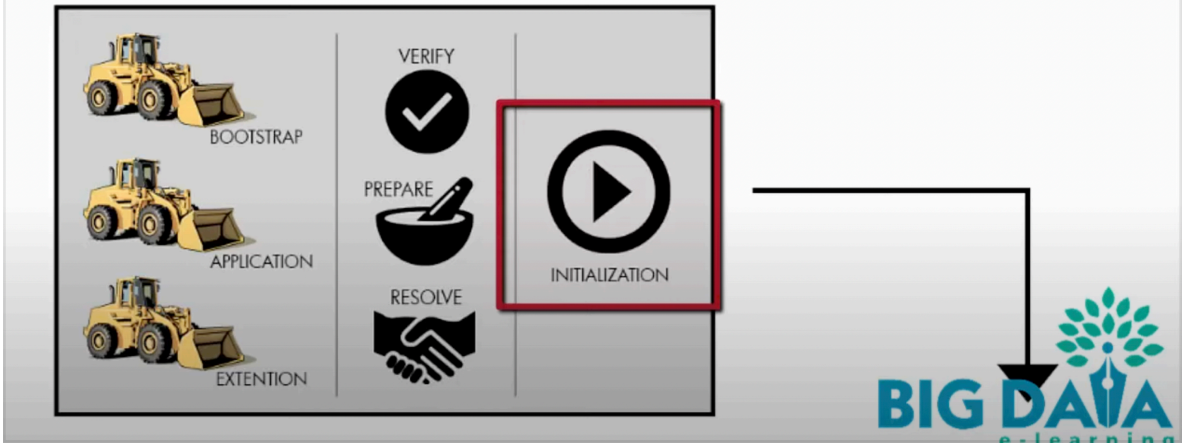


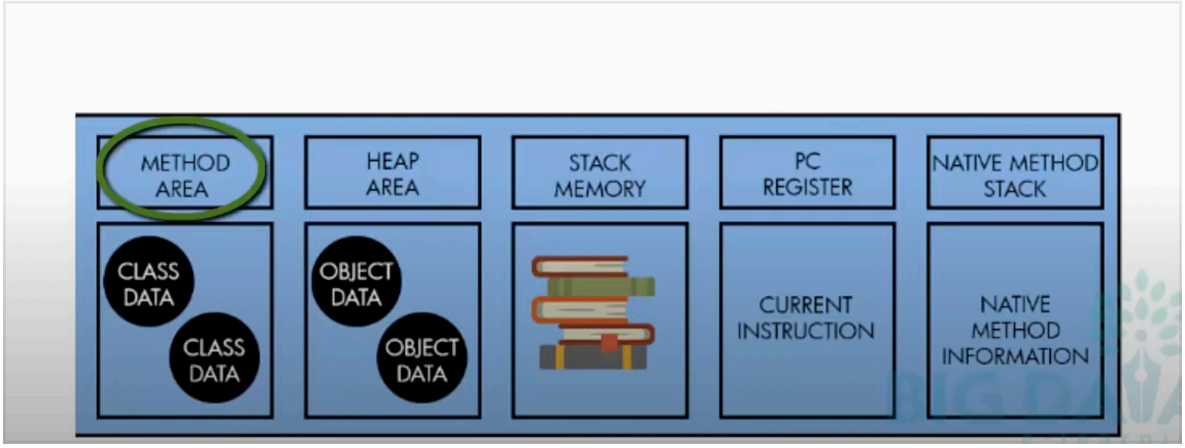


Linking

- **Verify**
  Once the class files are loaded to the memory, there is a verify phase where the bytecode class files are verified if they conform to standards
- **Prepare**
  In prepare phase, memory is allocated for the static variables and default values are assigned.
- **Resolve**
  In resolve phase, all the symbolic references are replaced with actual references.

In the "Initialization phase" of class loader component, all the static variables are assigned with the actual values and it also executes the static initializers at this point of time

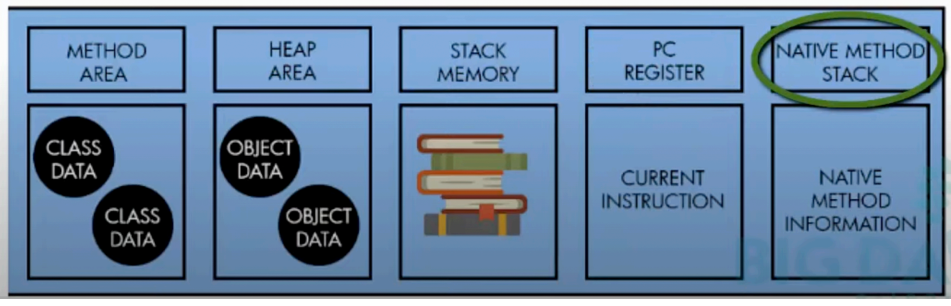




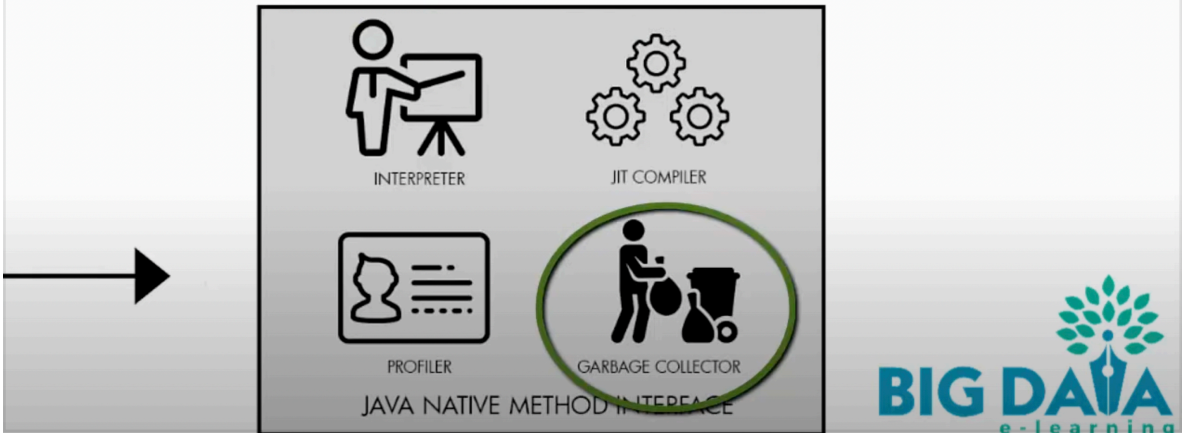
- **Method area**
  All the class level data are stored in this memory area.
- **Heap memory**
  All the objects and instance variables are stored.
- **Stack memory**
  > Local variable
  > Operand stack
  > Frame data
- **PC Register**
  This holds the current executing instructions.
- **Native Method Stacks**
  This holds the native method information.

```
Class Employee
{
static int count=0;
..
}
```

new Employee()



- The 3rd main component in JVM architecture is the execution engine. This is the actual engine that converts the bytecode to machine code and executes the instructions.



- Class loader
  - loading phase,
  - linking phase, and
  - Initialization phase.
- Run time data area
  - method area,
  - heap memory,
  - stack memory,
  - PC registers &
  - JNI.
- Execution engine
  - Interpreter,
  - JIT compiler,
  - JNI, and
  - Garbage collector.