Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

SAIRAM DIGITAL RESOURCES

YEAR II SEM III

CS8392

OBJECT ORIENTED PROGRAMMING
(Common to CSE, EEE, EIE, ICE, IT)

UNIT NO 1

INTRODUCTION TO OOP AND JAVA FUNDAMENTALS

1.3 The Java Environment ,
Java Source File Structure ,
Compilation.

COMPUTER SCIENCE & ENGINEERING

The Java Environment : JVM

What is JVM?

JVM (Java Virtual Machine) executes java bytecode line by line.

JVM provides runtime environment in which java byte code can be executed.

JVMs are available for many hardware and software platforms - **JVM is platform dependent.**

Operations of JVM are:

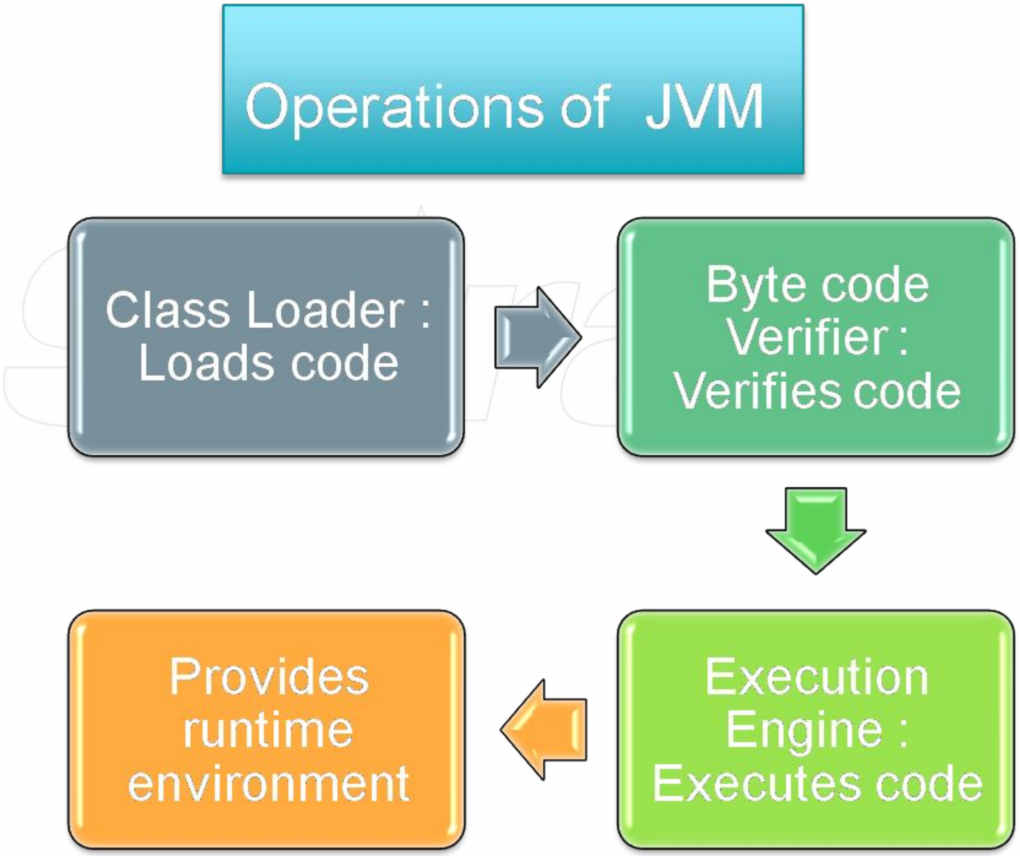
Class Loader : Loads code

Byte code Verifier : Verifies code

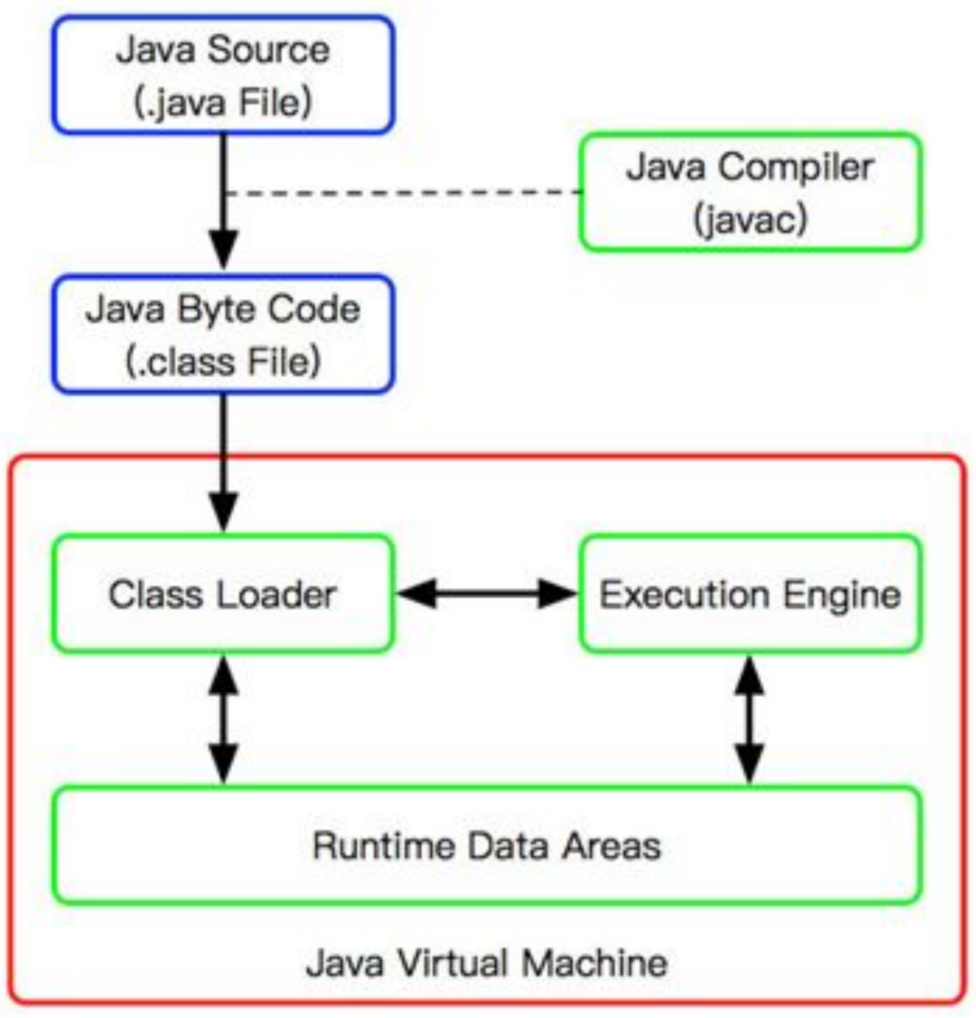
Execution Engine : Executes code

Runtime Data Area: Provides runtime environment

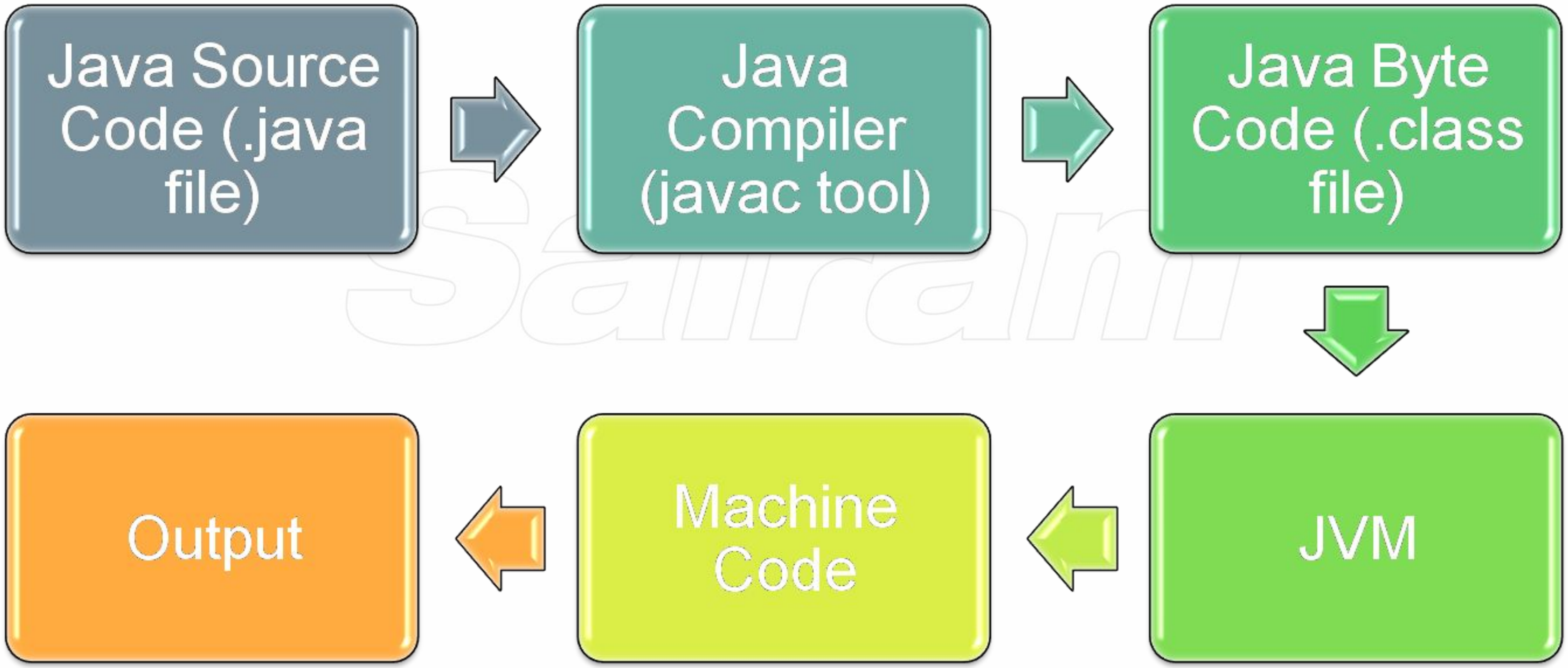
The Java Environment : JVM


Operations of JVM
Class Loader : Loads code
Byte code Verifier : Verifies code
Execution Engine : Executes code
Provides runtime environment

The Java Environment : JVM



The Java Environment : JVM


Java Source Code (.java file)
Java Compiler (javac tool)
Java Byte Code (.class file)
JVM
Machine Code
Output

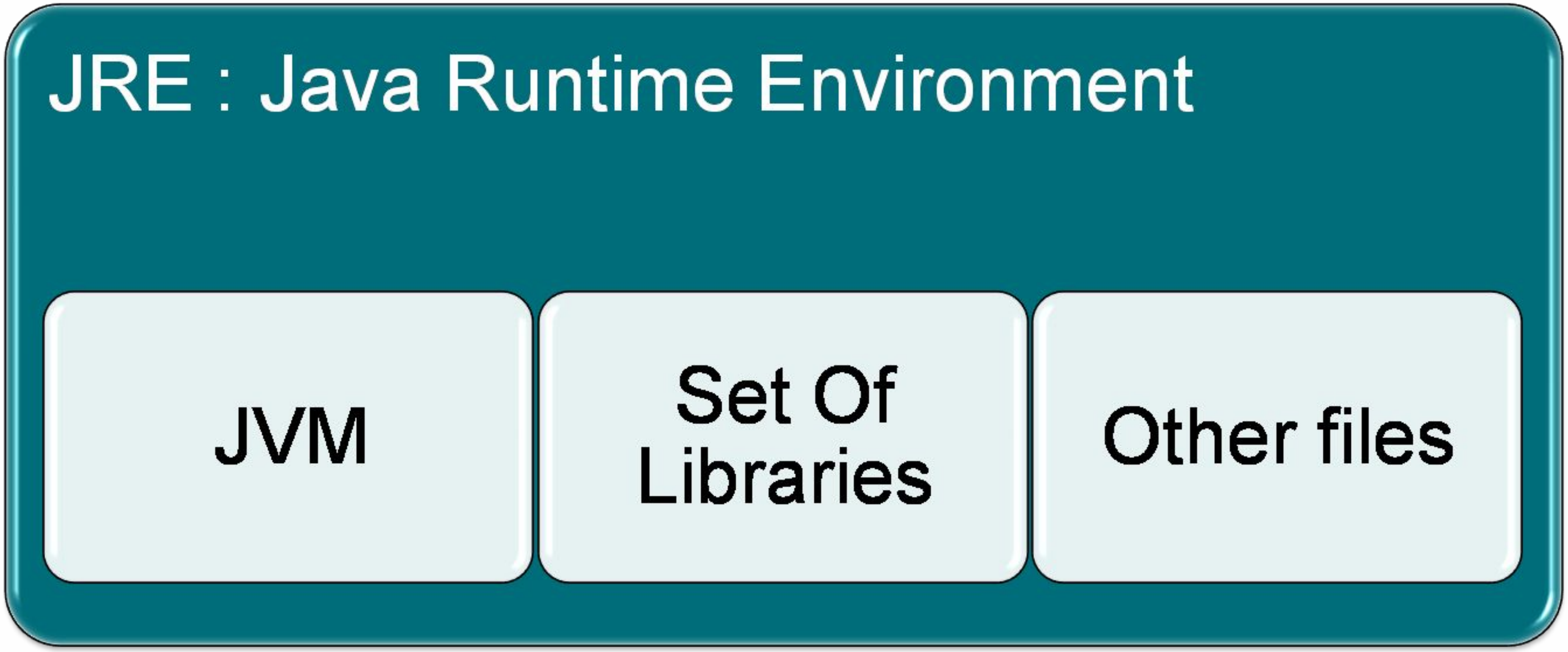
The Java Environment : JRE

- Java Environment refers to JRE (Java Runtime Environment).
- JRE is the implementation of JVM
- JRE is minimum requirement to run any java code..
- Components of JRE are JVM, set of libraries (used at runtime) and other files.

- **JRE = JVM + Libraries + Other files**

The Java Environment : JRE

JRE = JVM + Libraries + Other files



The Java Environment : JDK

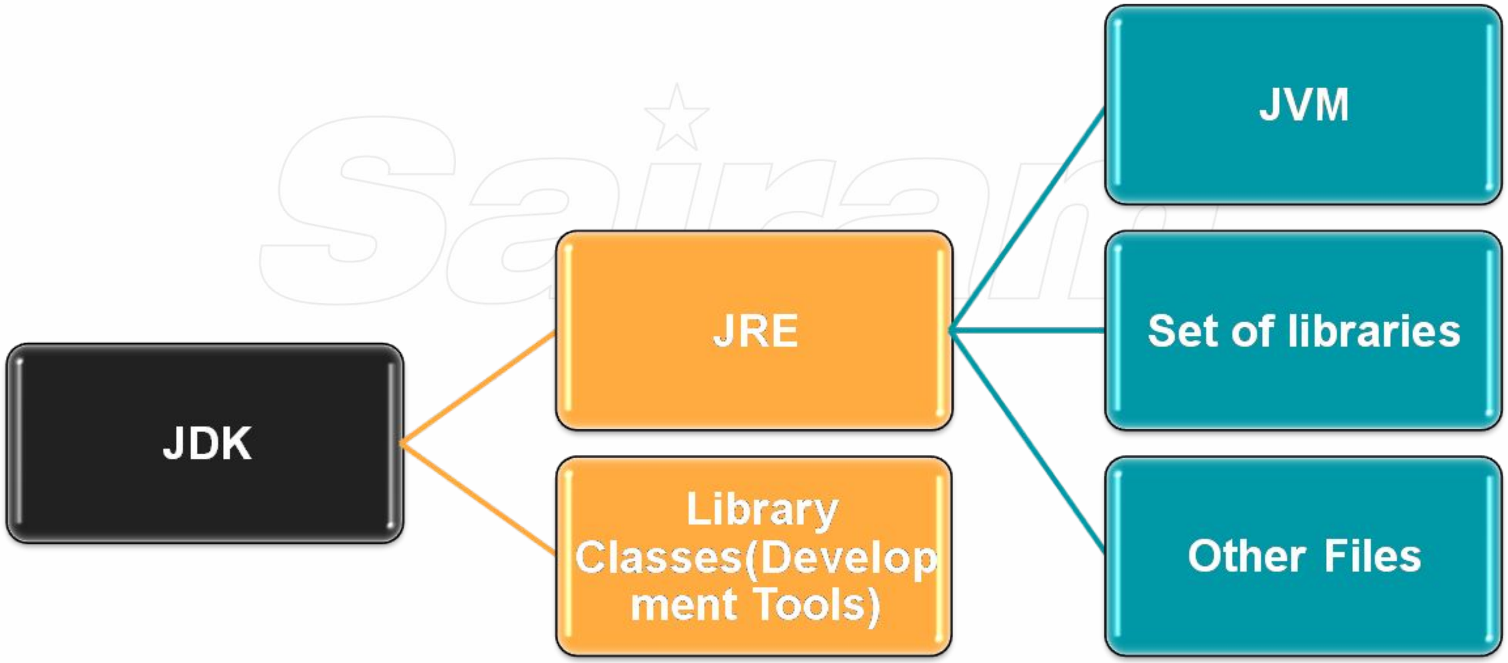
- Java Development Kit (JDK) is a software development environment used to develop java applications .
- It physically exists.
- JDK is an implementation of Java Platform by Oracle corporation(J2SE, J2EE and J2ME)

- **JDK = JRE + Library Classes (development tools)**

- **Development tools are debugger , compiler ,javaDoc etc**

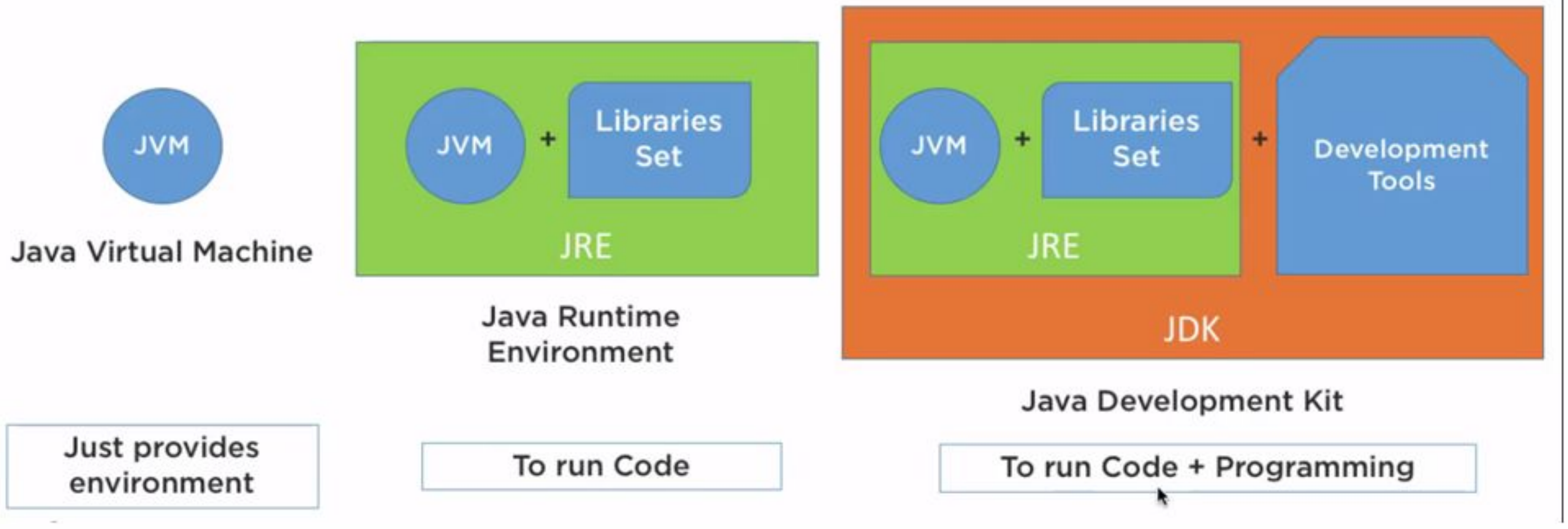
The Java Environment : JDK

JDK = JRE + Library Classes (development tools)
JRE = JVM + Libraries + Other files



The Java Environment : JDK

JDK = JRE + Library Classes (development tools)
JRE = JVM + Libraries + Other files

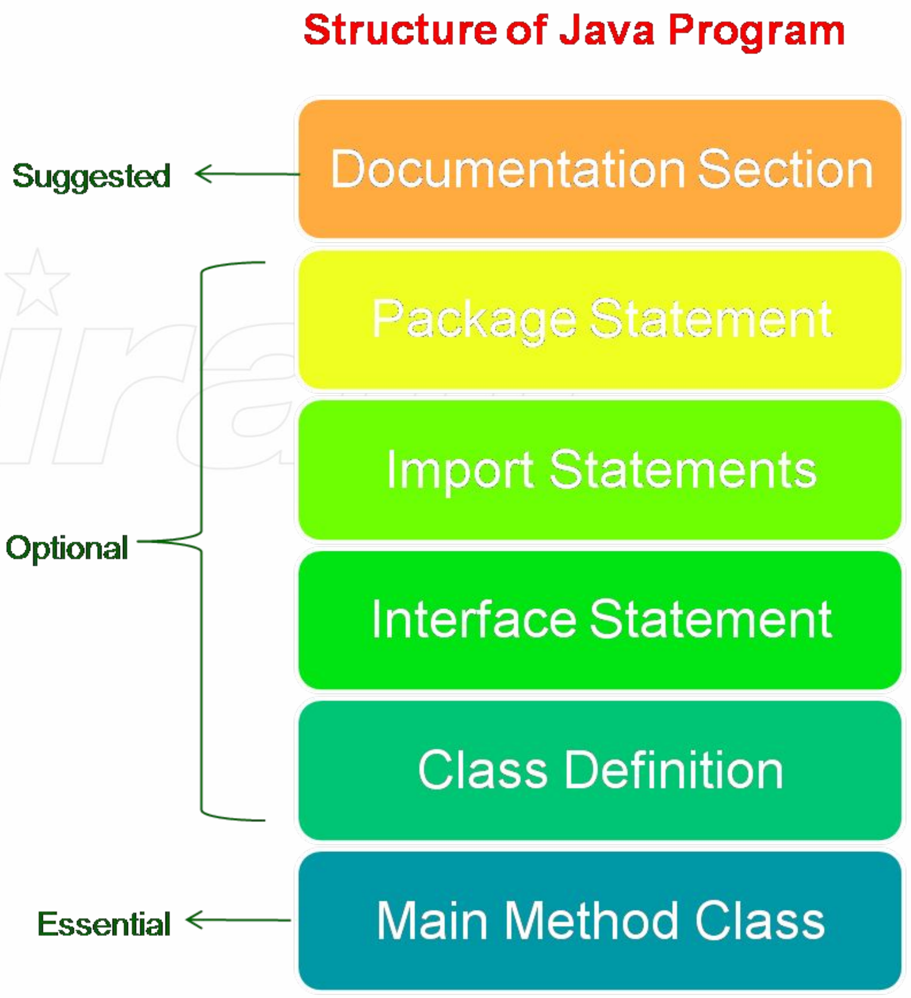


Java Source File & Structure

General structure of java program will be:

- Documentation Section
- Package Statement
- Import Statements
- Interface Statement
- Class Definition
- Main Method Class

    -- Main Method Definition



Java Source File & Structure

Documentation Section

- Written by comments. It helps to understand the code. And additional info about program.
- It is an optional part of the program,

```
/* a documentation comment
         starts with a delimiter and ends with */
```

Package Statement

- A package is a group of classes that are defined by a name.
- There can be only one package statement in a Java program and it has to be at the beginning of the code .
- It is an optional part of the program,

Syntax:

```
package package_name;
```

Example :

```
package sairam_cse;
```

Java Source File & Structure

Import Statements

- To use a class of another package we can do it by importing.
- An import statement is must be written after the package statement.

Syntax:

Import package_name:

Example

import java.util.Date; //imports the date class 1

Java Source File & Structure

Interface Statement

- Interface is used to specify an interface in Java..
- An interface is a similar to a class in Java but it contains only constants and method declarations.

Syntax::

```
interface interface_name
{
   /* All the methods are public abstract by default
     As you see they have no body     */
   public void method();
}
```

Java Source File & Structure

Interface Example:

```
interface cse_inter
{
   public void method1();
   public void method2();
}  // end of interface
class Demo implements cse_inter
{
     public void method1()
   {
   System.out.println("implementation of
method1");
 }
public void method2()
   {
System.out.println("implementation of
method2");
}

 public static void main(String arg[])
   {
         Demo obj = new Demo();
         obj.method1();
   }  //main method
}  // end of Demo class
```

Java Source File & Structure abstract class and interface.

Ø Abstract class and interface both are used to achieve abstraction.
Ø Abstract class and interface both can't be instantiated.

Abstract class	Interface
abstract keyword is used to declare abstract class.	**interface keyword** is used to declare interface.
can **have abstract and non-abstract** methods.	can have **only abstract** methods.
doesn't support multiple inheritance.	**supports multiple inheritance**.
abstract class can be extended using keyword "extends".	**interface** can be implemented using keyword "implements".
class members can be private, protected, public,default	interface members are public by default.

Java Source File & Structure

Class Definition

- A class is a collection of variables and methods .
- Every program in Java will have at least one class with the main method.
- class is declared by use of the **class** keyword

Class Definition Syntax:

```
class classname {
        data type variable1;
        data type variable2;
        ...
        data type variableN;
        return type method_name1(parameter-list) {
                // body of method }
        ...
        type method_nameN(parameter-list) {
                // body of method  }
} // end of class
```

Java Source File & Structure

class example:

```
class cse
{
          int id;         //field or data member or instance variable
          String name;
           public void show()
                  {
                    id=5;
                  name="SureshAnand";
                  System.out.println(id);  System.out.println(name);
                  }
          public static void main(String args[])
                   {
                   cse s1=new cse();//creating an object of cse
                   s1.show();
                   }   // END OF MAIN METHOD
 }    // END OF CLASS
```

Java Source File & Structure

Main Method Class

- There may be many classes in a Java program but **only one class defines the main method**.

Example:

```
public class cse   // public : accessible from any other classes
{
    public static void main(String[] args)
    {
        System.out.println("This is my First Java Program");
    }
}
```

Java Source File & Structure

Main Method Definition: ***public static void main***

***public* main method** : it can be outside of its class; accessed by all

static main method : access a method without creating its object . ie.before creating any class objects

void : main method does not return a value

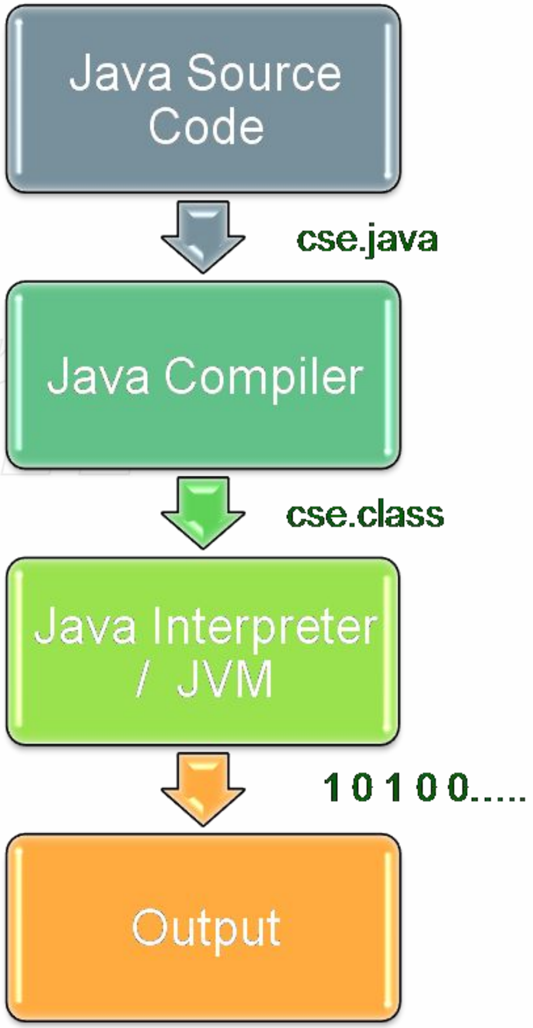
main method: starting point of a Java program.

String[] args : each element passed as input array of string ; everything as string

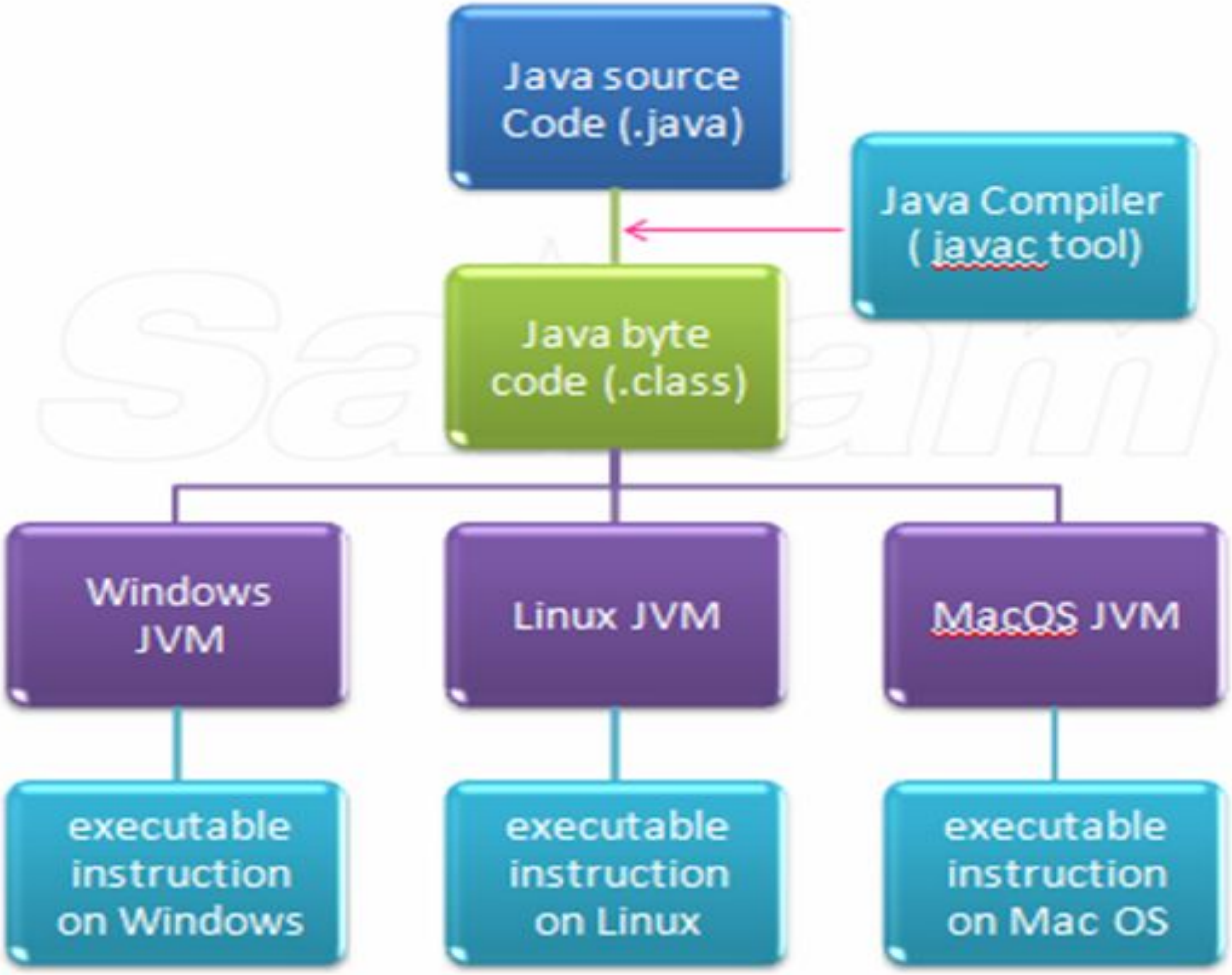
Compilation

Compilation :

- Java program is compiled by javac tool.
- Compiler generates bytecode (.class file).
- Then byte code executed by interpreter (JVM).
- i.e byte code converted as machine code by java interpreter.



Compilation


Java source Code (.java)
Java Compiler (javac tool)
Java byte code (.class)
Windows JVM
Linux JVM
MacOS JVM
executable instruction on Windows
executable instruction on Linux
executable instruction on Mac OS

Compilation

- To compile java program we have to set path.
- We are not using IDE like eclipse, netbeans etc.

Setting Temporary java Path in Windows : Steps to set the temporary path java

1. Open the command prompt
2. Copy the path of the JDK/bin directory
3. Write in command prompt:
4. set path= jdk path upto bin

Example:

```
set  path =  C:\Program Files\Java\jdk1.8.0_40\bin;
```

Compilation

Another way path : Setting permanent java Path in Windows

Navigate as follows

MyComputer -> properties -> advanced system setting -> advanced tab -> Select environment variables -> click new tab of user variable

```
Variable name= path
Variable value= C:\Program Files\Java\jdk1.8.0_40\bin;
```

Then give ok.

Compilation

Open notepad or any text editor type java program and save as ***file_name.java***.

To compile:

open command prompt navigate to java file location

```
javac file_name.java
```

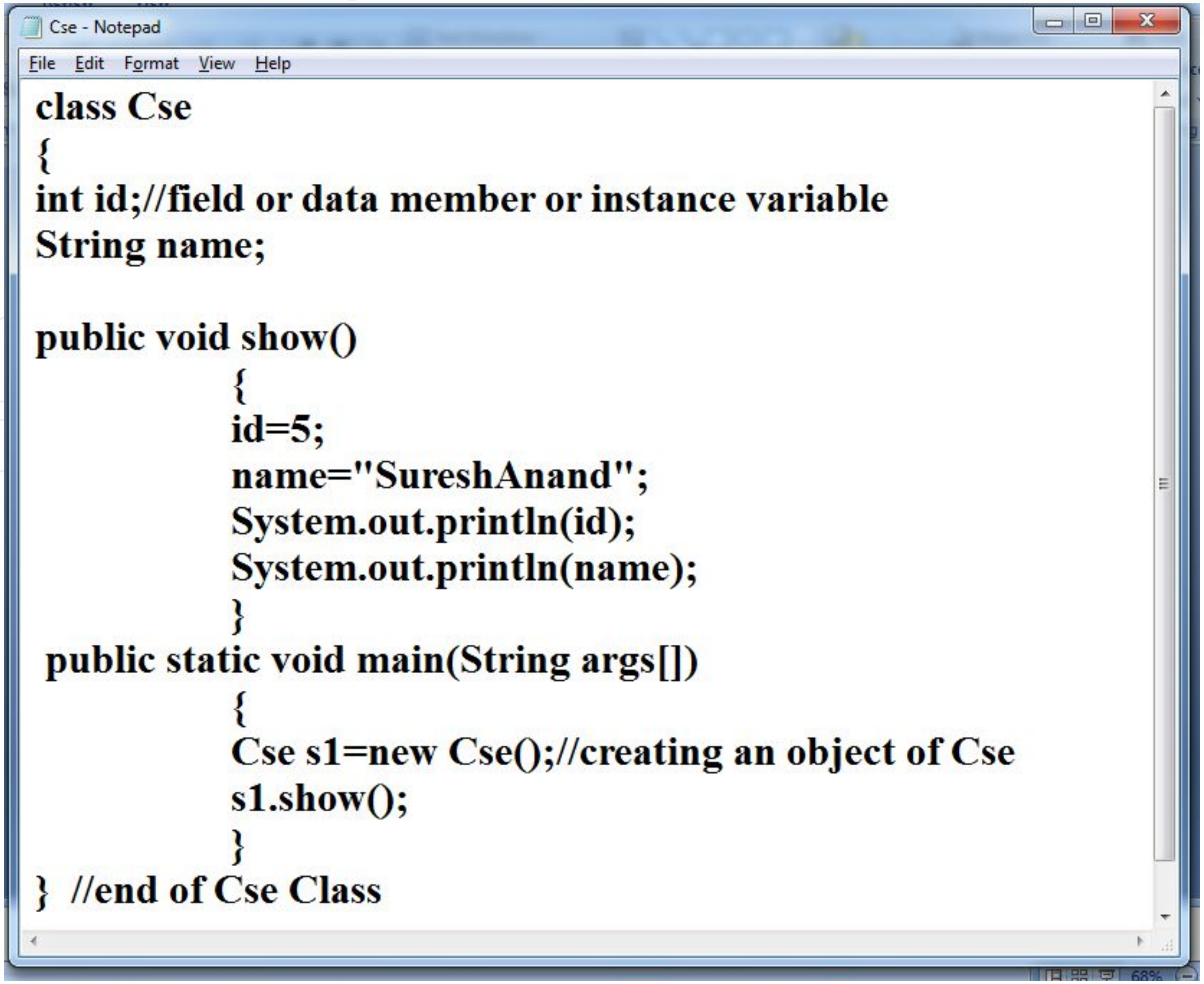
Now class file will be created. (i.e byte code)

To execute:

```
java file_name
```

java interpreter executes byte code into machine code(instructions) .i.e generates output.

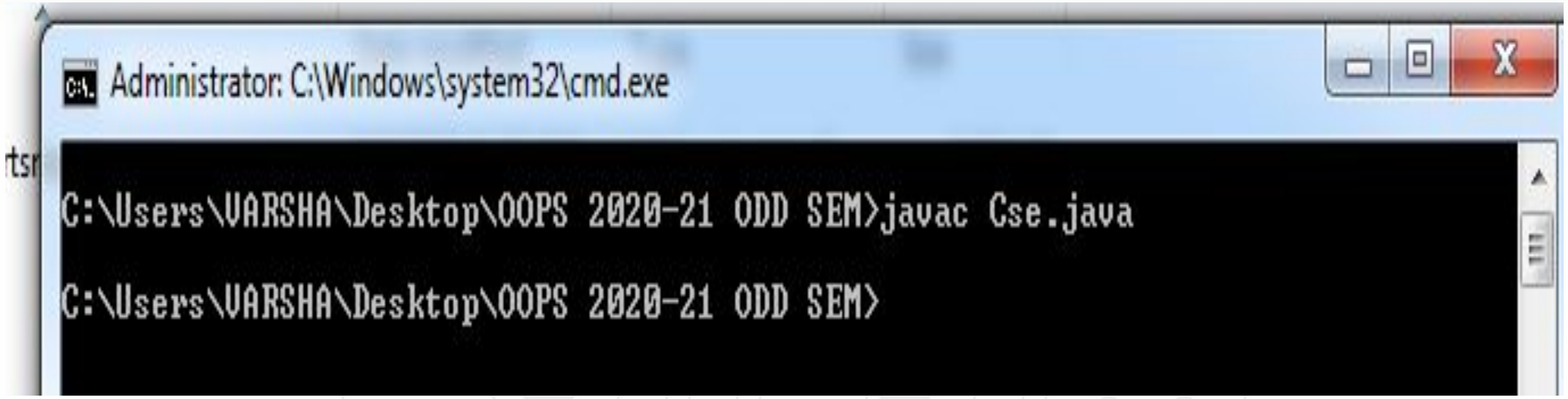
Compilation



```
class Cse
{
int id;//field or data member or instance variable
String name;

public void show()
		{
		id=5;
		name="SureshAnand";
		System.out.println(id);
		System.out.println(name);
		}
 public static void main(String args[])
		{
		Cse s1=new Cse();//creating an object of Cse
		s1.show();
		}
}  //end of Cse Class
```

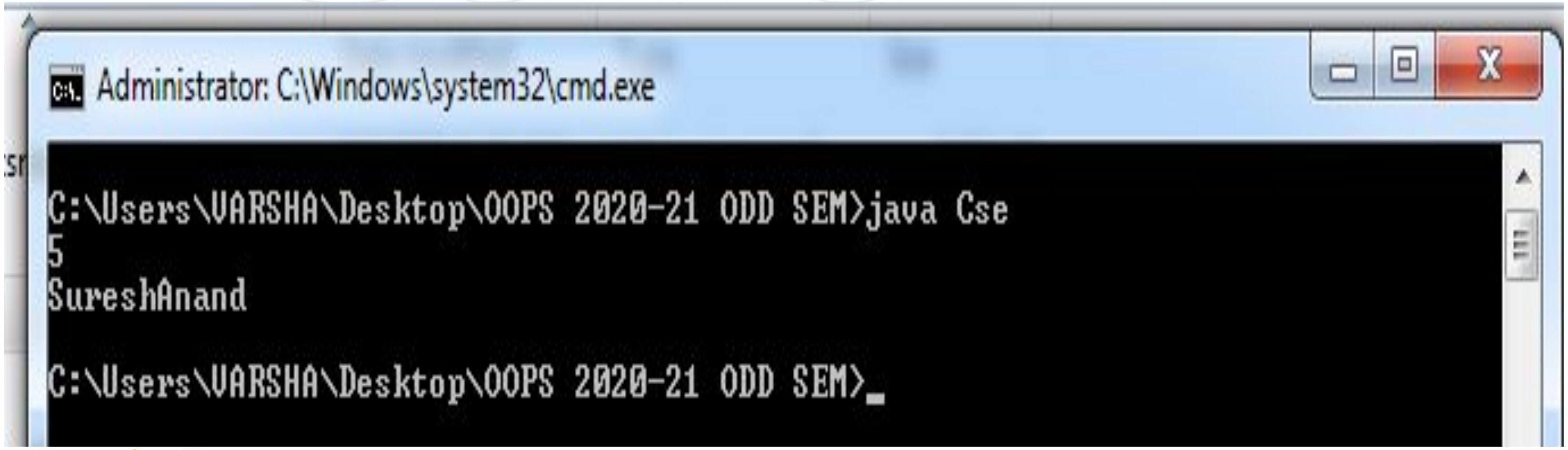
Compilation



```
Administrator: C:\Windows\system32\cmd.exe

C:\Users\VARSHA\Desktop\OOPS 2020-21 ODD SEM>javac Cse.java

C:\Users\VARSHA\Desktop\OOPS 2020-21 ODD SEM>
```



```
Administrator: C:\Windows\system32\cmd.exe

C:\Users\VARSHA\Desktop\OOPS 2020-21 ODD SEM>java Cse
5
SureshAnand

C:\Users\VARSHA\Desktop\OOPS 2020-21 ODD SEM>_
```

Variants of Print statement

- Consider this print statement:

  System.out.println("We will not use 'Hello world!' ");

  This method displays the string parameter on the console. It then terminates the output line so that each call to println displays its output on a new line.

- Even if a method takes no parameters, you must still use empty parentheses.

  System.out.println();

  For example, there is a variant of the println method with no parameters that just prints a blank line.

- There also is a print method in System.out that doesn't add a new line character to the output.

  For example, System.out.print("hello");

  System.out.print("one");

  prints helloone without a new line. The next output appears immediately after the "o".

Compilation

Few online java compilers for students:

1.Codiva.

2.JDoodle

3.Rextester..

4.OnlineGDB..

5.Browxy. .

6.IDEOne.

Compilation

Few IDE java programming:

1.Eclipse IDE.

2.NetBeans.

3.BlueJ.

4.IntelliJ IDEA.

5.jEdit

6.DrJava.

7.jCreator.

8.Android Studio.

Video Link

https://youtu.be/T0Aeu_nmqE0

https://youtu.be/rxtk6jfSFGg

https://youtu.be/Ra6GeFpjBBE

https://youtu.be/fhfVkPpIwjk