Java Core Day 1

What is java?

Java is a programming language. It is used to develop desktop application, Web application, mobile application and so many other application development. It works on the concept of oops -
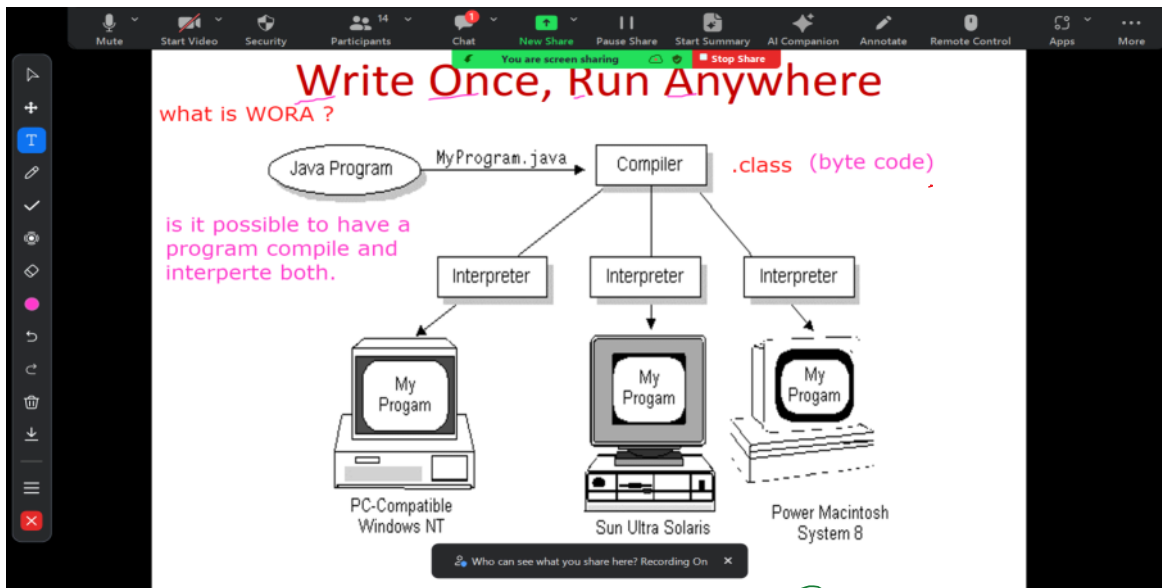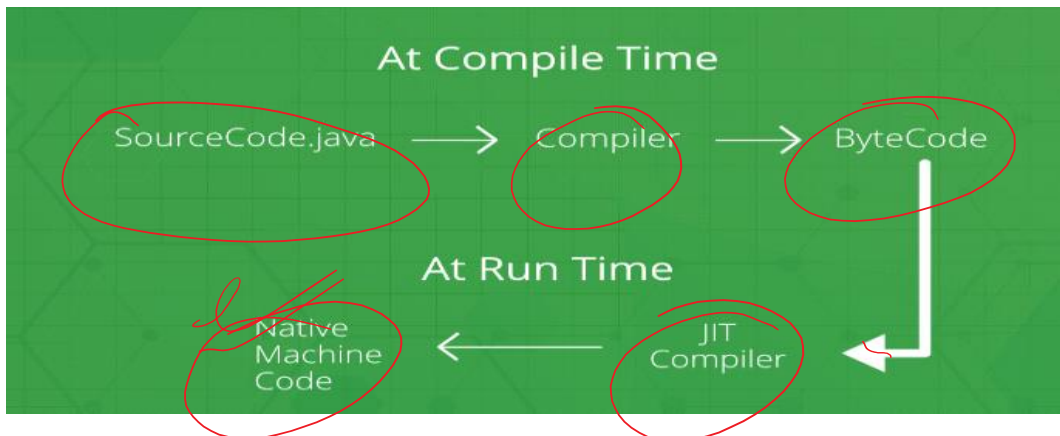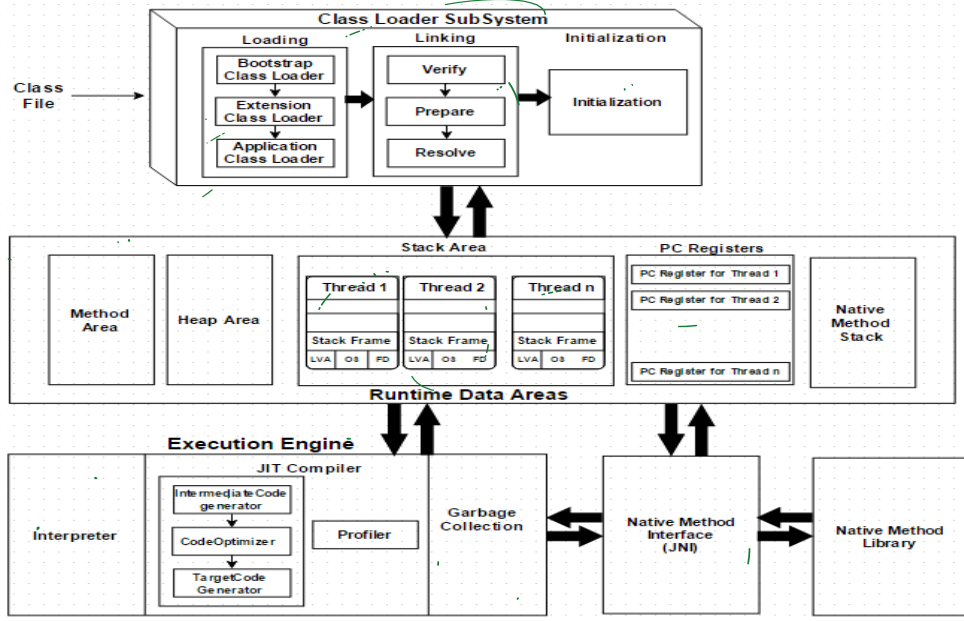The concept of OOPS are -

1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism
5. Class
6. Object

- Java is an object oriented programming language. It has syntax similar to C and C++ languages, but has vastly improved featur es.
- Java is platform independent.
- Java platform provides a complete environment for application development on desktop and servers and for deployment in embedd ed environment.
- Java is a **Programming language** and **a platform.**
- Java is a high level programming language
- **Platform -** Any hardware or software environment in which a program runs, in known as a platform. Since java has a runtime environment(JR E) and API, it is called a platform.
- **API(java application interface)** - is a list of all classes that are the part of the JDK. It includes all java packages, classes, interfaces along with their fields, methods and constructor.
- A complete listing of all classes in JavaAPI can be found at oracle website:

    http://docs.oracle.com/javase/7/docs/api

- These prewritten classes provide a tremendous amount of functionality to a programmer.
- A programmer should be aware of these classes & should know how to use them.

Types(Flavor) of Java Applications -
There are mainly 4 types of applications that can be created using java programming : -

1. Standalone Application   - J2SE - Java 2 Standard Edition
2. Web Application
3. Enterprise Application
4. Mobile Application   J2ME - JAVA 2 MOBILE EDITION

J2EE - JAVA 2 ENTERPRISE EDITION

Class Loader SubSystem

Loading
- Bootstrap Class Loader
- Extension Class Loader
- Application Class Loader

Linking
- Verify
- Prepare
- Resolve

Initialization
- Initialization

Class File

Runtime Data Areas

Method Area | **Heap Area**

Stack Area
- Thread 1 — Stack Frame — LVA OS FD
- Thread 2 — Stack Frame — LVA OS FD
- Thread n — Stack Frame — LVA OS FD

PC Registers
- PC Register for Thread 1
- PC Register for Thread 2
- PC Register for Thread n

Native Method Stack

Execution Engine

Interpreter

JIT Compiler
- Intermediate Code generator
- CodeOptimizer
- TargetCode Generator

Profiler

Garbage Collection

Native Method Interface (JNI)

Native Method Library

At Compile Time

SourceCode.java → Compiler → ByteCode

At Run Time

Native Machine Code ← JIT Compiler ← ByteCode

Write Once, Run Anywhere

what is WORA ?

Java Program → MyProgram.java → Compiler → .class (byte code)

is it possible to have a program compile and interperte both.

- Interpreter → My Progam → PC-Compatible Windows NT
- Interpreter → My Progam → Sun Ultra Solaris
- Interpreter → My Progam → Power Macintosh System 8

```
public static void main(String args[]){


}
```

System.out.println("____?");

class keyword to declare a class — **public class Demo** — Name of User defined class — String is a class

{

Opening of class — Return type(not specific)

public static void main(String args[]) — Object of String Class

Access Modifier {

name of method

Keyword(it is use for self initilization)

Opening of Main

System.out.println("Hello welcome to java");

system is a class

}

Access Operator

Terminator

}

method argument

Object of Print Stream class

Method(Overridden)

Closing of main method

Closing of class

Core Java Page 3