POP -> OOP
OOP -> Functional Programming

Java
1. Prerequisit of CPP
2. Start with Java Directly on Abstract level

OOSD
1. OOA -> Requirements in terms of Object
2. OOD -> Designing of objects
3. OOP -> Java

```
Employee{
id
name
sal
}
```

```
Student{
int rollno
string name
double marks
}
```

```
Date{
int day
int month
int year
}
```

It has its own Rules
It has its Own Syntax
It overcomes the limitations of other languges

OOP
1. major pillars
   - Abstraction (Object Creation, Function call)
   - Encapsulation (class, defining a function)
   - Modularity (Dividing the code into smaller modules)
   - Hirerachy (Reusability)
     - has-a Relationship(Association)
     - is-a RelationShip (Inheritance)
2. minor pillars
   - Typing/ Polymorphism (Single entity that can take multiple forms)
     - Complie Time Polymorphism (Function Overloading)
     - RunTime Polymorphism (Function Overriding)
   - Concurrency
     - Multithreading
   - Persistance
     - File IO -> HardDisk(Local Machine)
     - JDBC -> Server

Java
It has its own syntax
It has its own rules

Java Platforms
1. Java card (Smart Cards)
2. Java ME (Smaller Devices)
3. Java SE (Standard Edition -> Desktop Applications)
4. Java EE (Enterprise Edition -> Web Applications)

1992 -> *7
WWW -> Hot Java Browser

Java Development Kit (JDK)
- tools + docs + JRE (Java Runtime Environment)
- tools + docs + rt.jar + JVM (Java Virtual Machine)

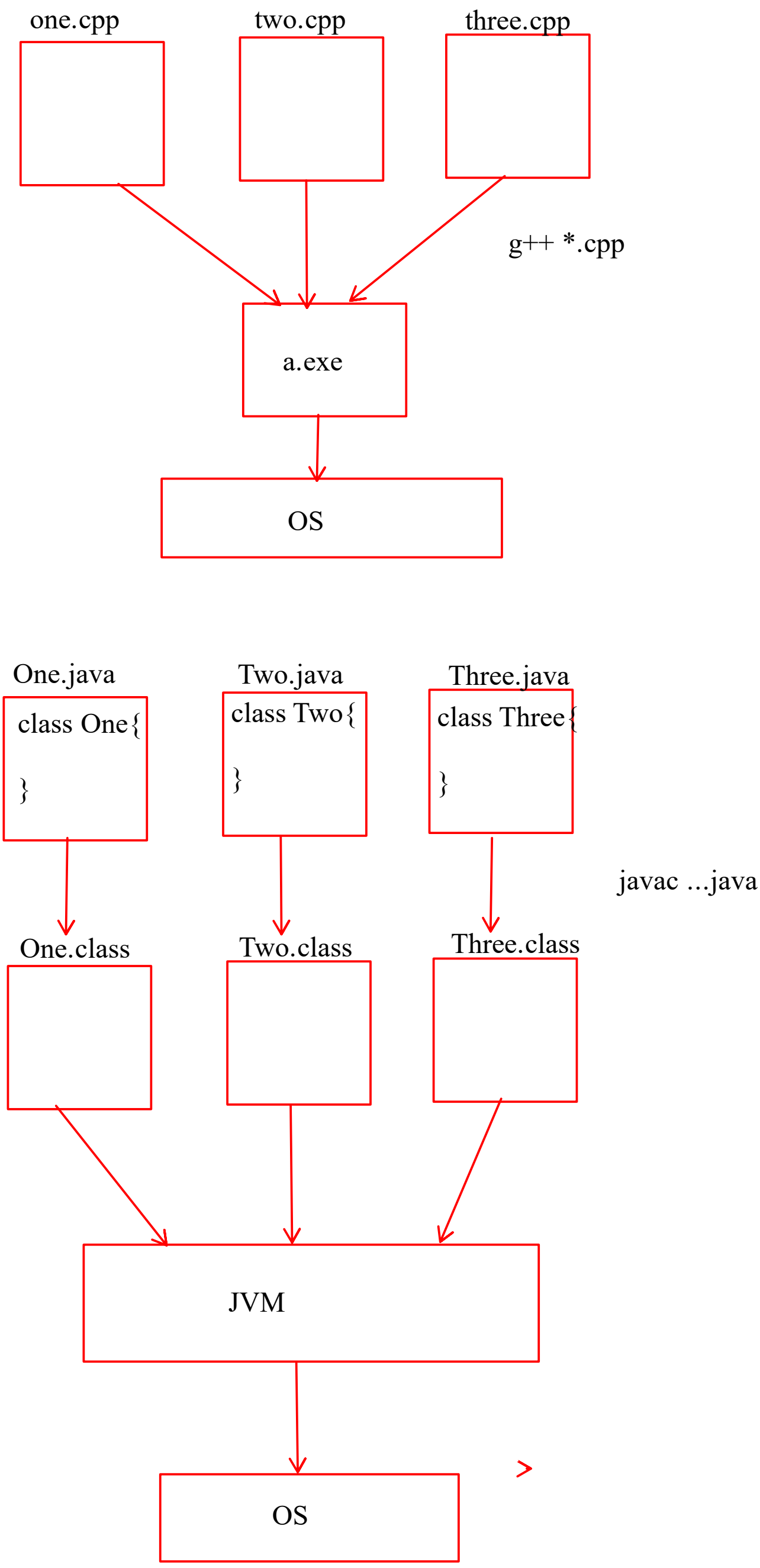
Compiler -> g++ , libraries

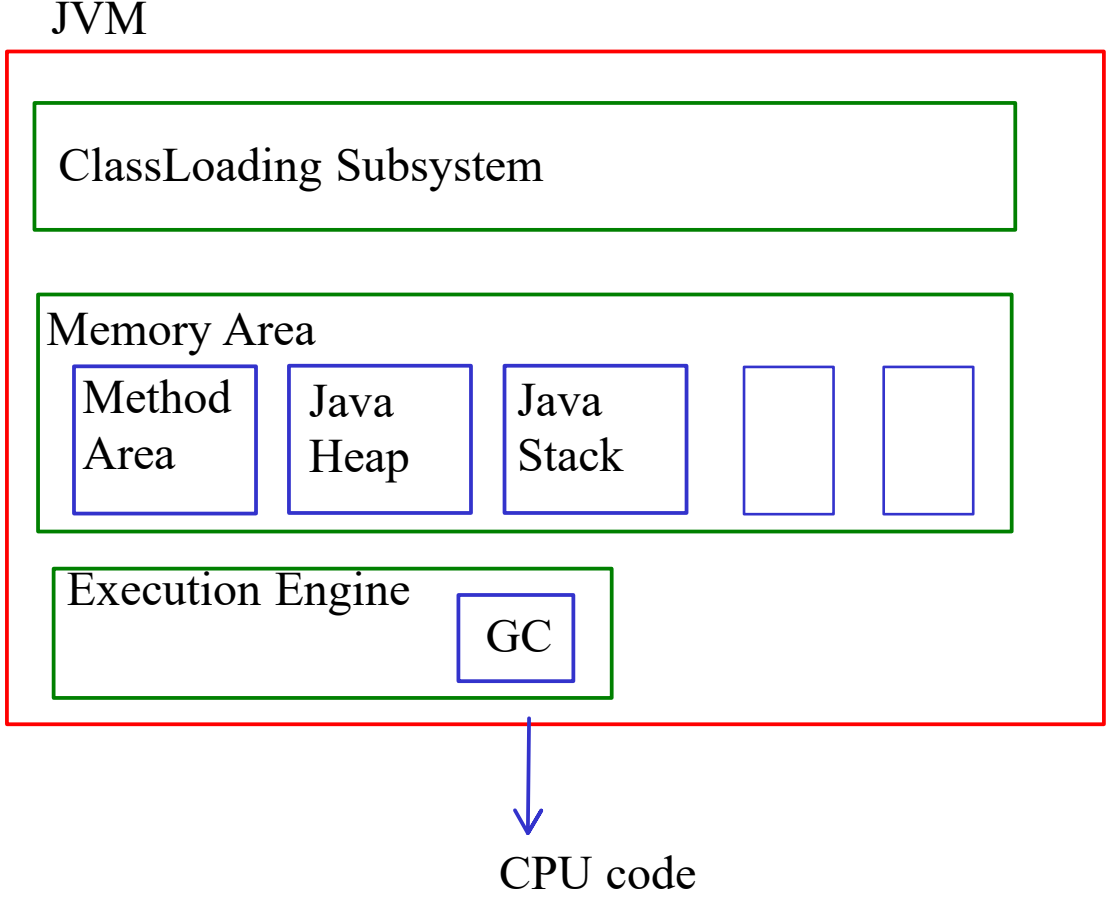
Program.java

```
javac
java
javap
jar
.
.
.
```

demo01.cpp -> a.exe -> OS
demo01.java -> .class -> JVM -> OS

Download From Local Server
JDK
IDE -> STS-3.9.18





Stack
Heap
Data
Text/code

Java Stack -> Local Variables
Java Heap -> Dynamic Objects
Method Area -> class,Methods,Static Fields

CLASSPATH

Workspace -> Area(Directory) in which we will hold our multiple projects

Classwork
-> Day01 , Day02, Day03

Assignments
-> Assignment01, Assignment02,....

Steps to cretae the project from command line
1. create a directory cmd_line inside which add 2 sub directories src and bin.
2. create Program.java into the src directory
3. Cretae a class withs ame name as that of file(Program.java)
4. Add a main method and hello world inside it.
5. Open the cmd prompt and go inside the src directory
   - cd cmd_line\src
6. compile the code
   - javac -d ..\bin Program.java
7. Set the classpath
   - SET CLASSPATH=..\bin
8. Exceute the code
   - java Program

Lab
- Installation
- Classwork
- commandline
- Documentation
- Overloading, Class, Object, class Pointer, this pointer,
- Diagram of classs object on stack and heap section