

Name - rohan paramane	Fundamentals	
Experiance - 8+ years	OOP	Collection Framework
	Functional programming	
	MultiThreading	
	Reflection	

Programming

OOSD - Object Oriented Software Development

1. OOA - Object Oriented Analysis
2. OOD - Obejct Oriented Design
3. OOP - Object Oriented Programming

1. OOA ->	Attendance System
	student
	employee
	attendance

2. OOD ->	student{	employee{	attendance{
	int rollno;	int empid;	Time intime
	String name;	String name;	Time outtime
	String address;	String address;	Date date
	String mobile;	String mobile;	
	}	}	}
	Time{	Date{	
	int hr;	int day;	
	int min;	int month;	
	}	int year;	
		}	

3. OOP
 - CPP,Java,Python,C# ,etc....

Java -> OOP

OOP ? -> Methodology

Abstraction, Encapsulation,Inheritance, Polmorphism

1. Major Pillar

- Abstraction
 - Encapsulation
 - Modularity
 - Hirerachy
- JAVA

2. Minor Pillar

- Typing/Polymorphism
- Concurrency
- Persistance

Abstraction

- Getting to know only essential things
- eg-> Function call, Objects

Encapsulation

- Binding the data and code together
- Abstraction is not possible without encapsulation
- Function defination, defining a class

15000 -> Mediatek
60MP
16MP
4000 MAP

```
void add(int n1, int n2){      add(10,20);  
//code  
int res1 = n1 + n2;  
  
}
```

Modularity->

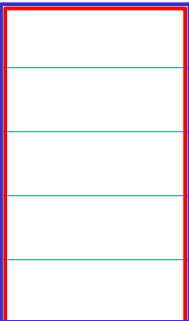
Dividing the code into smaller modules of differnt files

add(),sub()

printf("Hello World"); stdio.h
string string.h

- Hirerachy
- reusability (class)
 - has-a (Association)
 - is-a (Inhetitance)

Antenna -> Mobile device



Human has-a Heart
//dependent //dependency

Manager is-a Employee

Employee has-a Manager CPP,JAVA,Python,C#

Major Pillars

Minor Pillars

Polymorphism ->

- one entity that can take multiple forms
- Compile Time -> function/Method overloading
- RunTime -> function/Method Overrrding

Concurrency

- Performing/executing multiple tasks at the same time
- Mutithreading

Persistence

- Persisting/ storing the data permanantly
- File IO , JDBC

Java Platforms

- 1. Java Card -> Smart cards
- 2. Java ME -> Small Feature device application
- 3. Java SE -> Desktop Application
- 4. Java EE -> Web Application

Java SE	IDE
---------	-----

- JDK -> Java Development kit

JDK = tools + docs + JRE (Java Runtime Environment)

JRE = rt.jar + JVM(Java Virtual Machine)

rt.jar -> runtime jar -> all the core libraries of java

javac
javap
java
jar

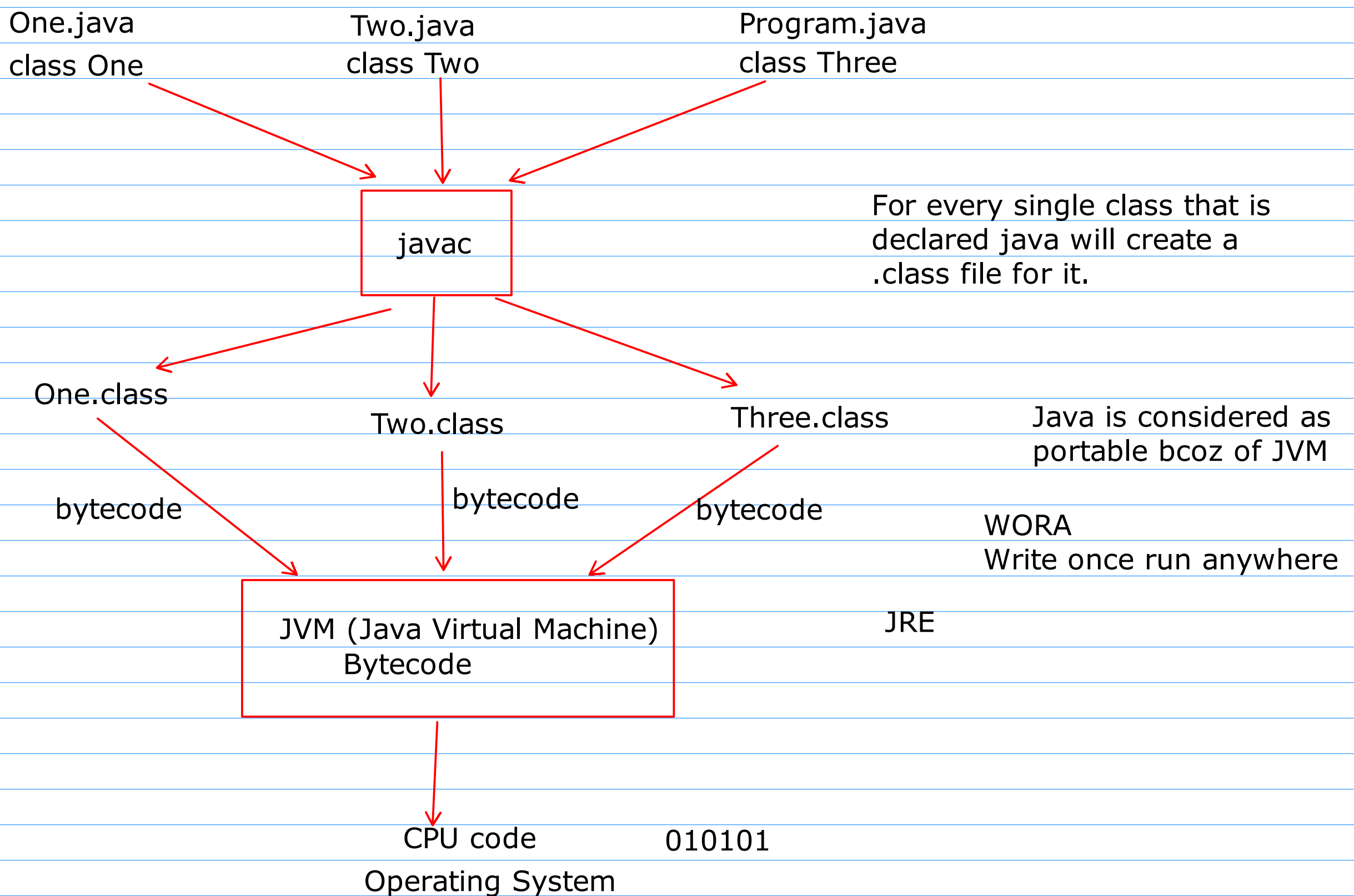
·
·
·
·

Hello World -> Terminal -> IDE

```
Program.java
// for compilation
    javac Program.java (name of .java file)
// after compilation
    Program.class
// for execution
    java Program (name of .class file to execute)
```

← this demo is optional

Flow of execution



q. How many `.class` files will be created?

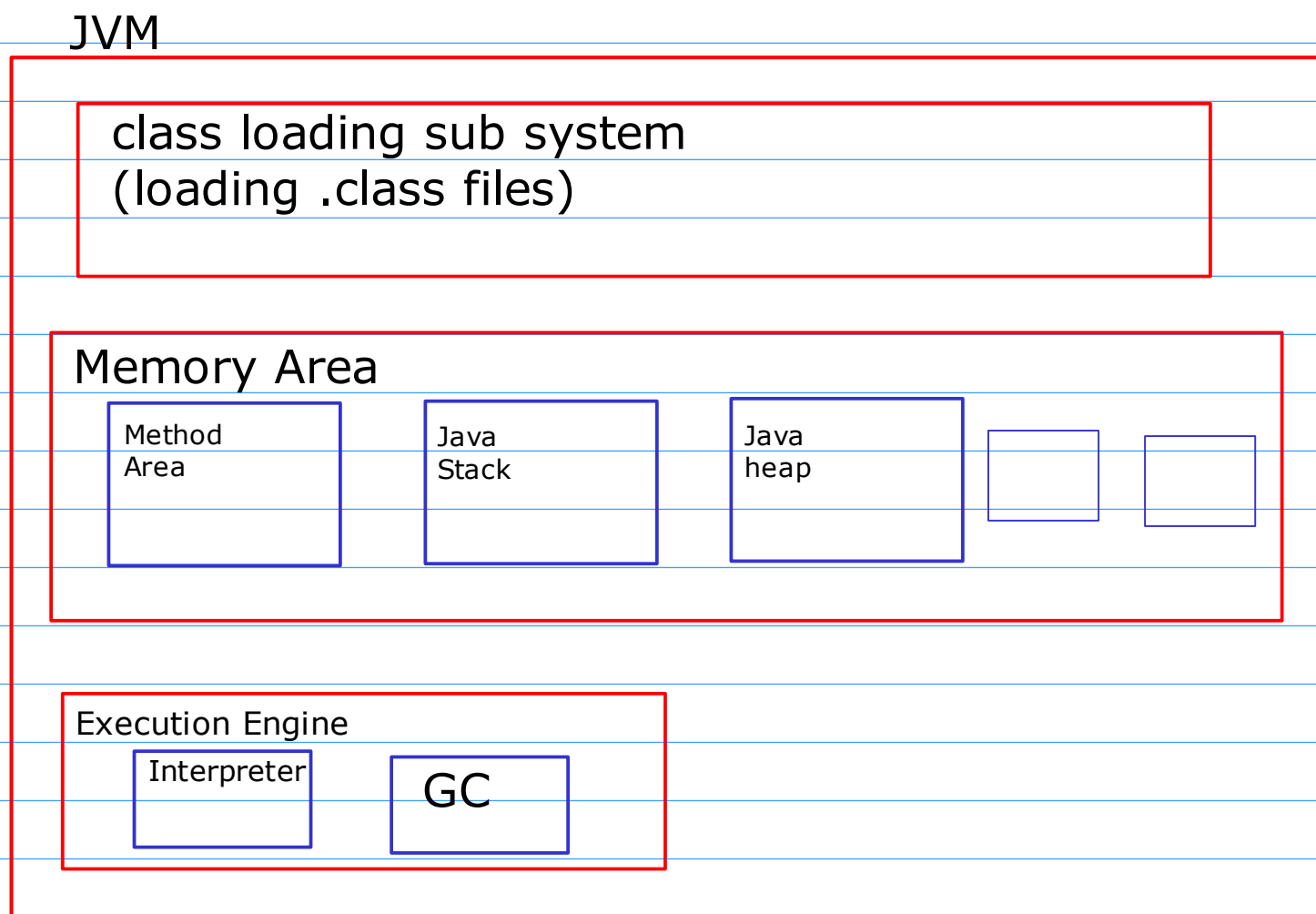
-> As many no of classes are created those many `.class` files will be created

q. How many `main`(entry point method) we can define in a single class?

-> only 1 `main`

q. how many `main`(entry point method) we can define in a single `.java` file

-> we can define `main` method in every single class. so as many class we want in single `.java` file we can define the `main` in them.



JVM Arch

Objects-> Dynamically GC

#class

- It is a logical entity
- It is also called as blueprint of an object
- It consists of data(fields) and code(methods)
- class is an example of Encapsulation

#Object

- It is a physical entity
- It is also called as an instance of the class
- Object is an example of abstraction

Q. how many public class we can declare in single .java file

- only 1

Q. name of public class and name of .java file should be different

-> false

-> It is the java language specification that the name of the .java file and name of public class should be same

hello world -> terminal compilation

public and non public class

class

Classpath