# Object Oriented Programming in Java

## Interview Questions

Q1) (a) What are the differences between Procedural Programming and object oriented Programming Languages?

(b) What are the drawbacks of procedural programming language?

(c) What are the needs/advantages of object oriented programming language?

# Procedural Programming

- Focuses on functions & logic!
- less reliable & secure
  - ↳ No access modifiers
- less reusability & difficulty in maintaining the codebase
- Example :→ C Programming language

Note:→ Javascript is functional programming language.

# Object oriented Programming

- Focuses on data & objects!
  - ↳ Data members & member functions
- More reliable & secure
  - ↳ Data hiding & Encapsulation
- More reusability & Easy maintainability
  - ↳ Due to Inheritance & Polymorphism
- Example :→ C++, Java, Python, C#

# Need of Object Oriented Programming

## PILLARS OF OOPS

### Encapsulation

- → classes & objects
- → Getters & Setters
- → Constructors (types & chaining)
- → this, static, final keyword

### Inheritance

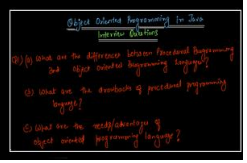
- → Types of Inheritance
- → IS A (Inheritance) vs has A (composition)
- → Supers keyword
- → Multiple Inheritance (Diamond Problem)

### Polymorphism

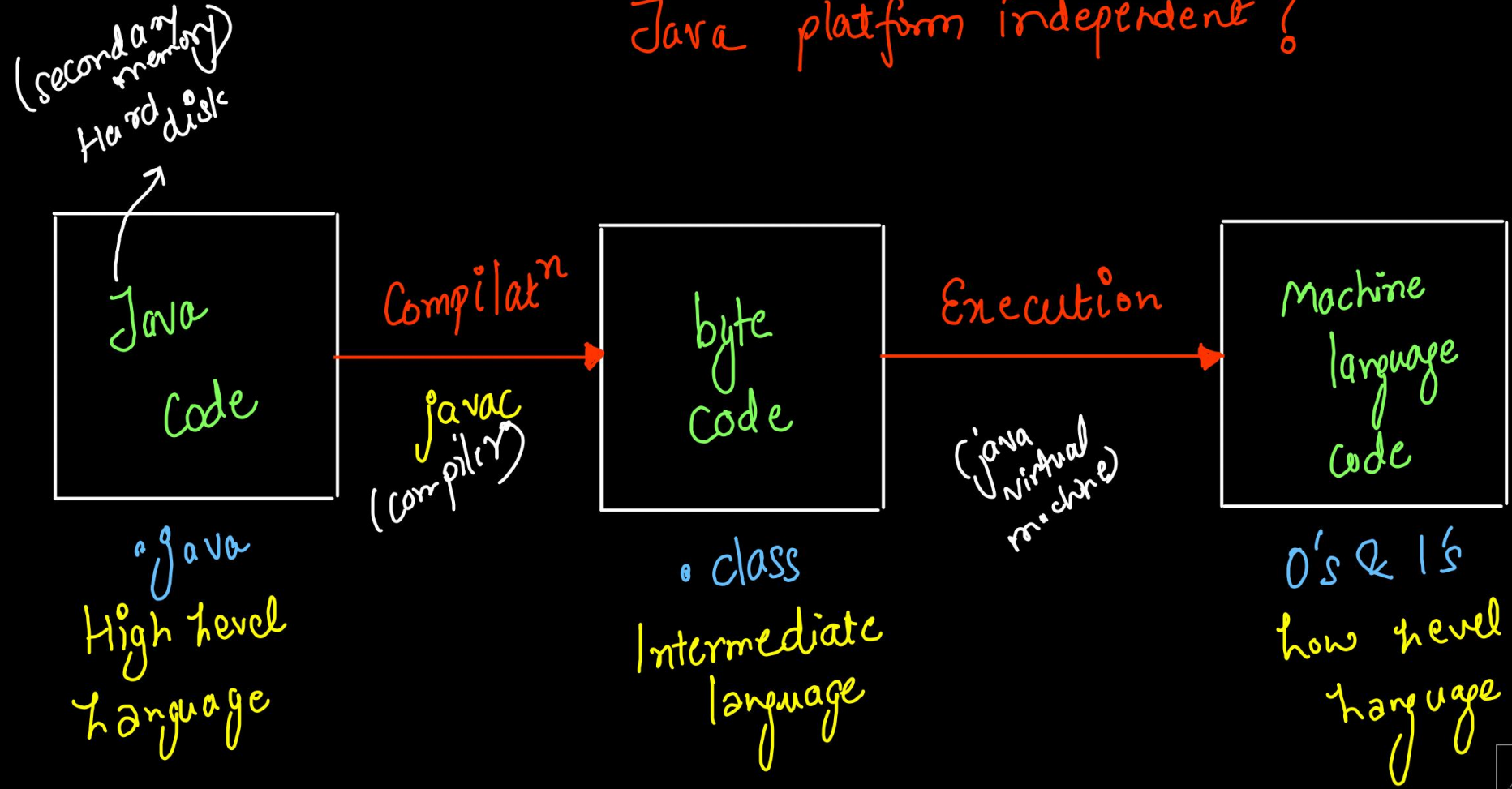
- → Compile Time or Static or method overloading
- → Runtime or Dynamic or method overriding

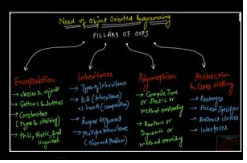
### Abstraction & Data Hiding

- → Packages
- → Access Specifier
- → Abstract classes
- → Interfaces



# Q) Explain Software Development Process in Java. What are various components in Java? Why is Java platform independent?



# Java Architecture

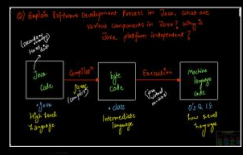
Java Development Kit ( JDK) = JRE + Development Tools

{ Interpreter, Class loader, compiler ( javac) }

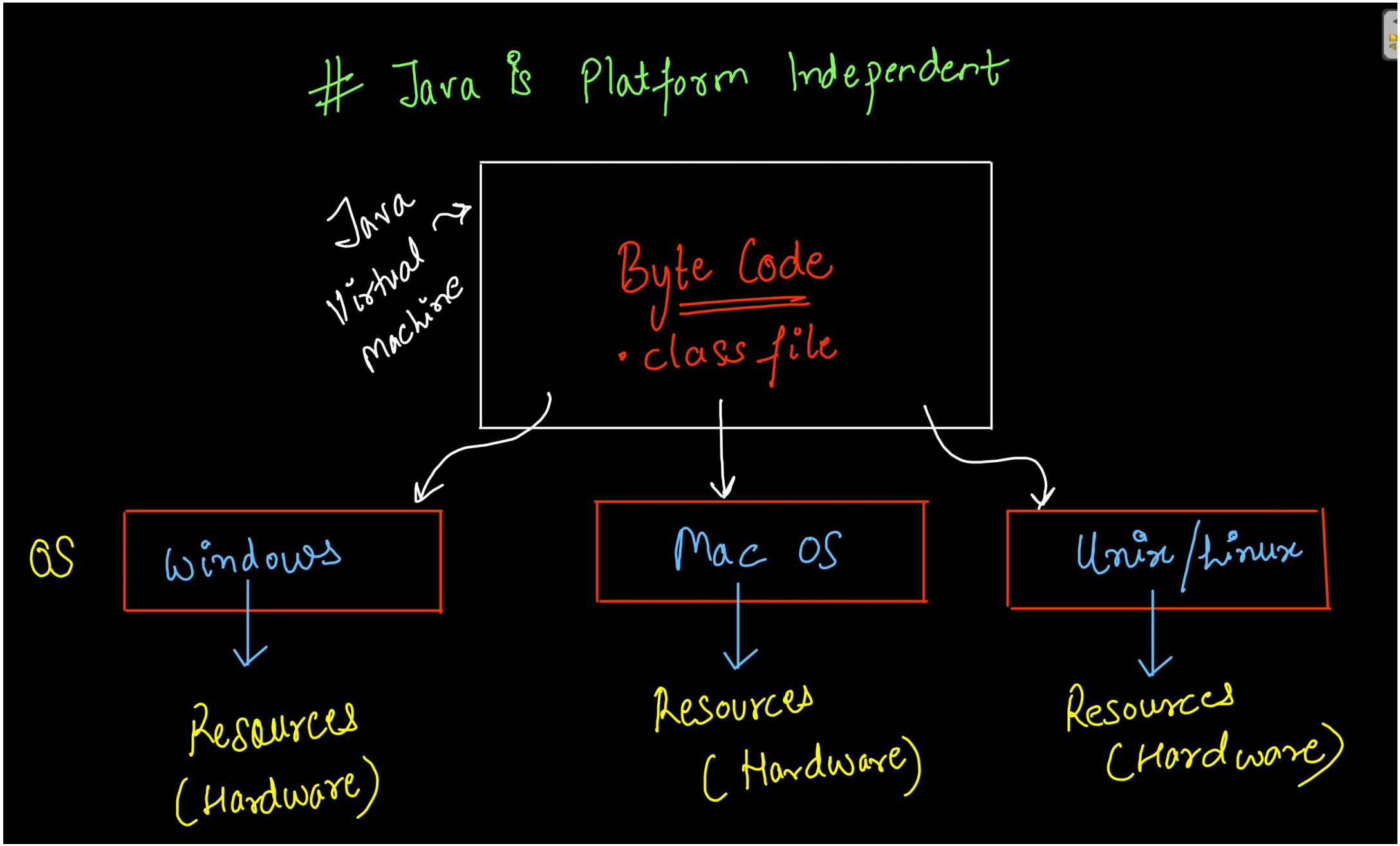
Java Runtime Environment ( JRE)

= JVM + library classes

Java Virtual Machine ( JVM)



# Java is Platform Independent



Q) What are differences between Program & Process? What do you mean by process memory layout? Explain the steps in process creation/ program execution?

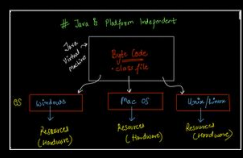
(secondary storage) Hard disk

# Program vs Process

# Program → Code/ Set of Instructions (.java)

# Program under execution is known as process

# Process = Program (Code) + Stack & Heap memory
+ Other resources
(CPU, disk, Network, I/O file, etc)



RAM

# Process Memory Layout (Random Access memory) main memory

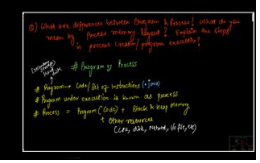
| Text / Code Segment | Function Call Stack | Heap Memory |
| --- | --- | --- |
| Program / Instructions | Primitive Variables<br>Reference Variables | Arrays, String literals,<br>Static variables,<br>Objects |

Heap Memory → scope = program (global); lifetime = program

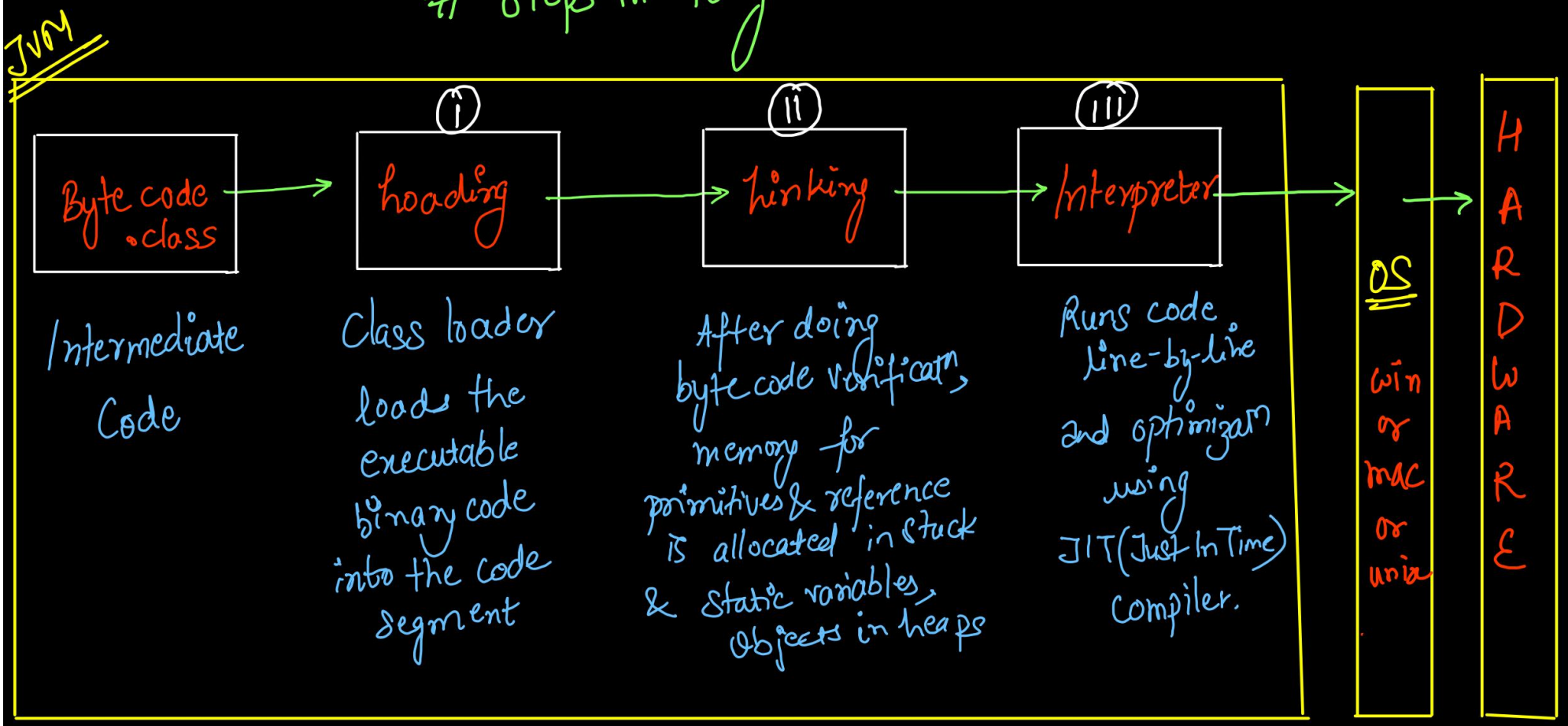
Function Call Stack → function scope, lifetime = fn

Primitive Variables: int, char, boolean, short, byte, long, float, double

| Global / Static Data Segment |
| --- |
| Static data, global data<br>Constants |



# Steps in Program Execution




Q) How are classes different from objects in Java? Explain with the help of real world example.

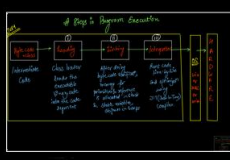
# Classes vs Objects

## Class

- Logical Entity
- do not occupy stack or heap memory space
- Blueprint for an Object
  - Data Members (Properties)
  - Member functions (Behavior)

## Object

- Physical / Real-world entity
- Occupies both stack & heap memory space
- Instance of a class
  - Reference Variable in stack
  - Actual Object in Heap



```java
class Movie{
    int duration;
    String name;
    double ratings;
    String genre;
}
```

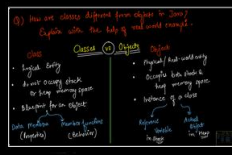
Class { Code → Text Segment }

```java
Movie avengers1 = new Movie();
avengers1.duration = 120;
avengers1.name = "Avengers Loki";
avengers1.ratings = 4.0;
avengers1.genre = "Action";

// System.out.println(avengers1);
System.out.println(avengers1.name + " " +
                   avengers1.duration + " " +
                   avengers1.genre + " " +
                   avengers1.ratings);
```

Object { Heap & Stack }

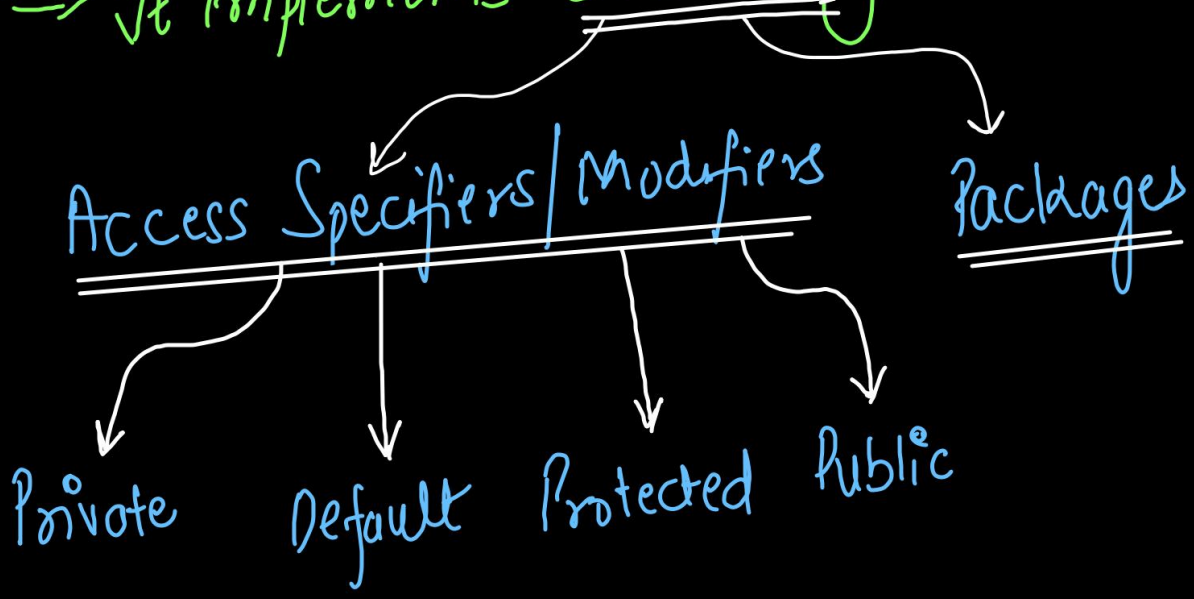
```
Avengers Loki 120 Action 4.0
```



Q) What is ENCAPSULATION? Explain it with the help of real world example!

"Wrapping together the data members and member functions into a single entity (known as class)"

⇒ It implements Data Hiding




```java
class Movie{
    private int duration;
    private String name;
    private double rating;
    private String genre;

    public void setDuration(int newDuration){ duration = newDuration; }
    public void setName(String newName){ name = newName; }
    public void setRating(double newRating){ rating = newRating; }
    public void setGenre(String newGenre){ genre = newGenre; }

    public int getDuration(){ return duration; }
    public String getName(){ return name; }
    public double getRating(){ return rating; }
    public String getGenre(){ return genre; }
}
```

properties (private)

setters

getters

```java
Movie avengers1 = new Movie();
avengers1.setDuration(120);
avengers1.setName("Avengers Loki");
avengers1.setRating(4.0);
avengers1.setGenre("Action");

System.out.println(avengers1.getName() + " "
                   + avengers1.getDuration() + " " +
                   avengers1.getGenre() + " " +
                   avengers1.getRating());
```