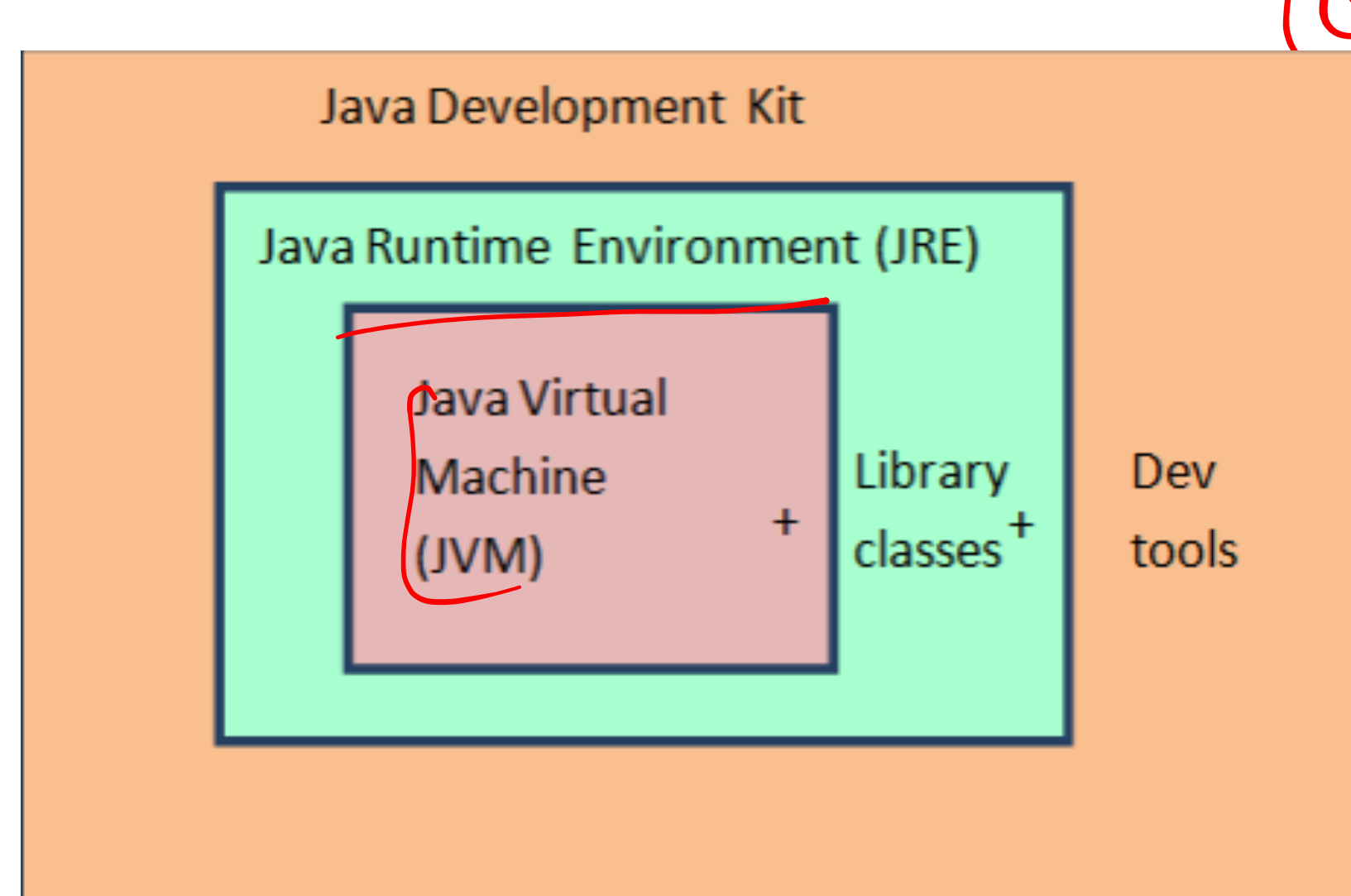
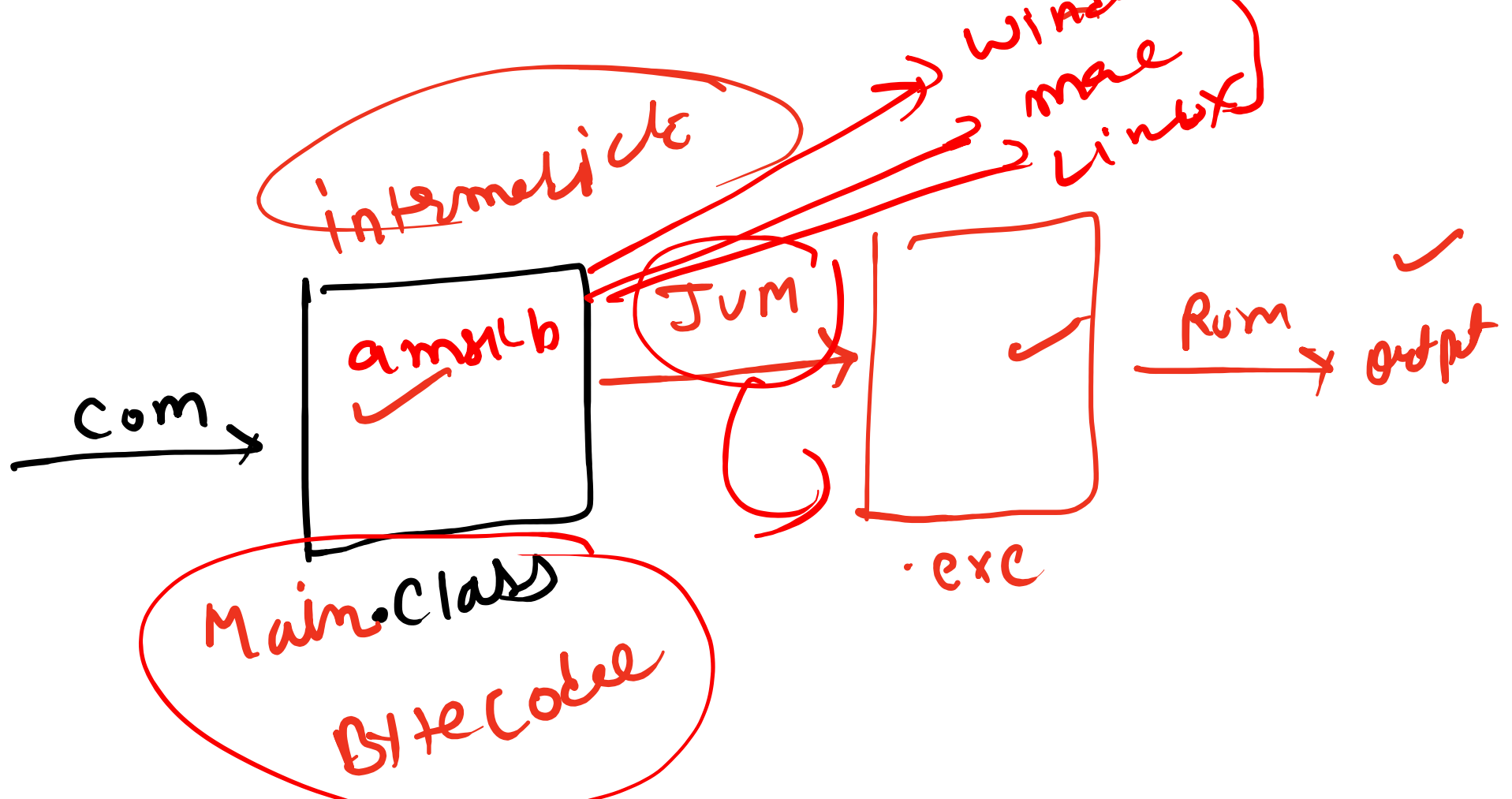
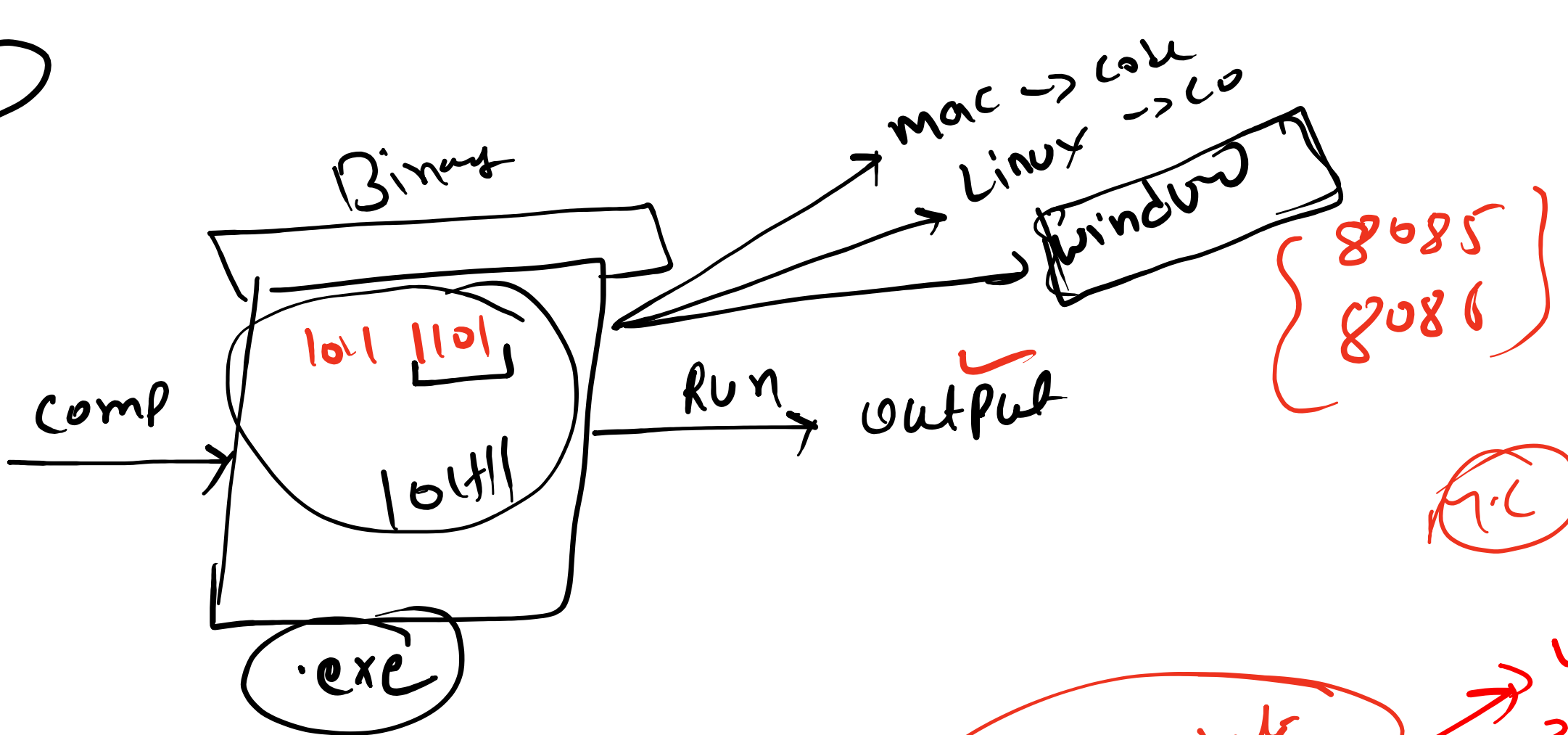


```
#include <iostream>
using namespace std;
int main() {
    cout<<"Hello World!";
}
```

.cpp

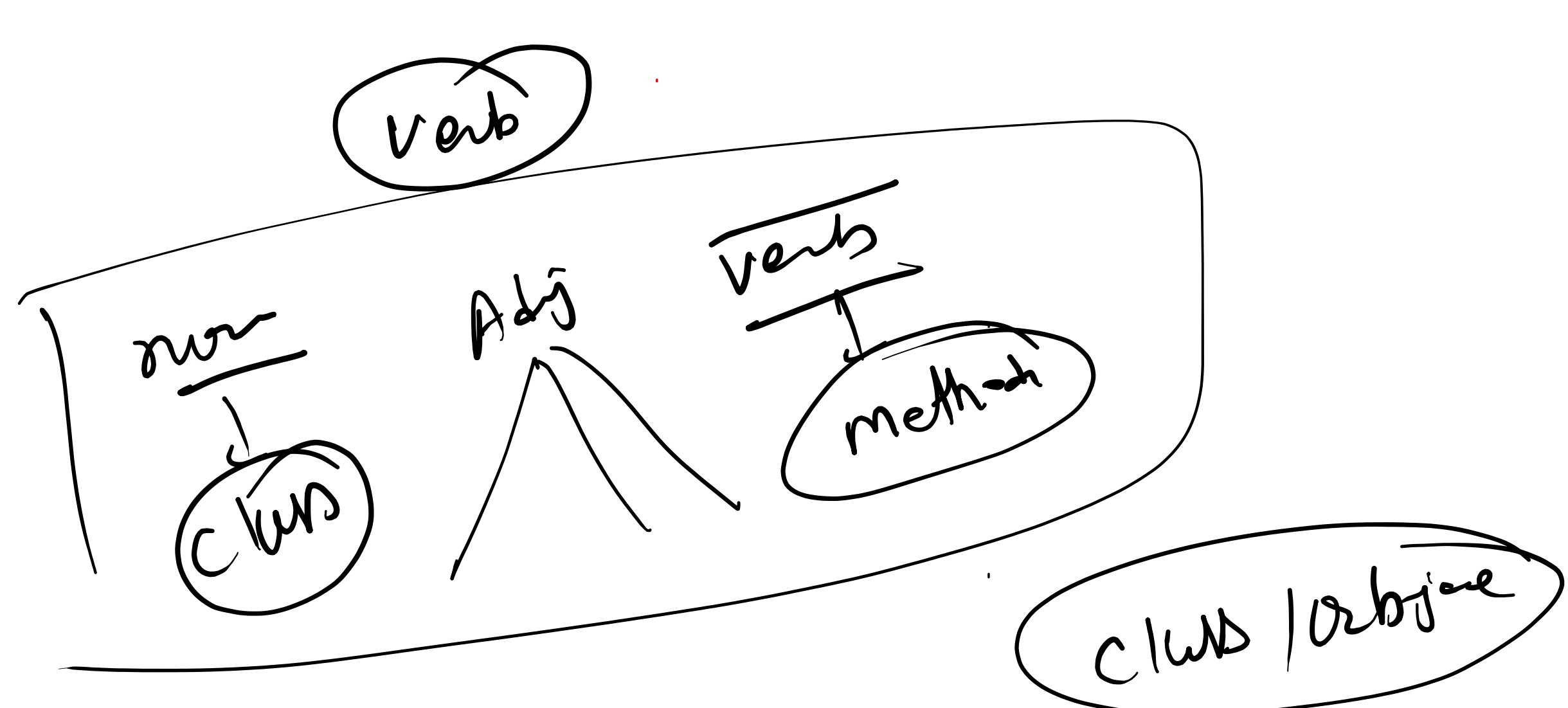
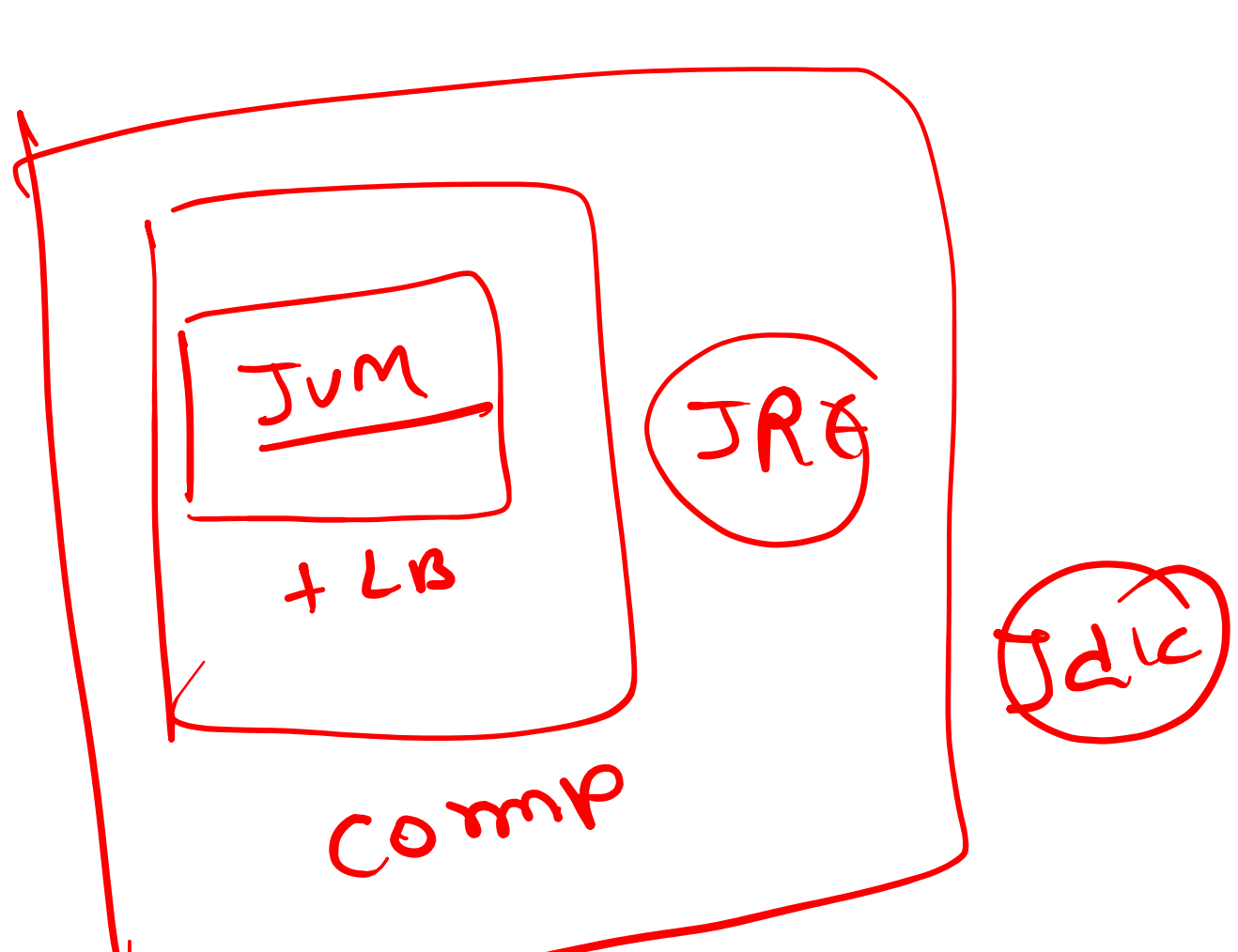
```
public class Main {
    public static void main(String args[]) {
        System.out.println("Hello World!");
    }
}
```

Main.java

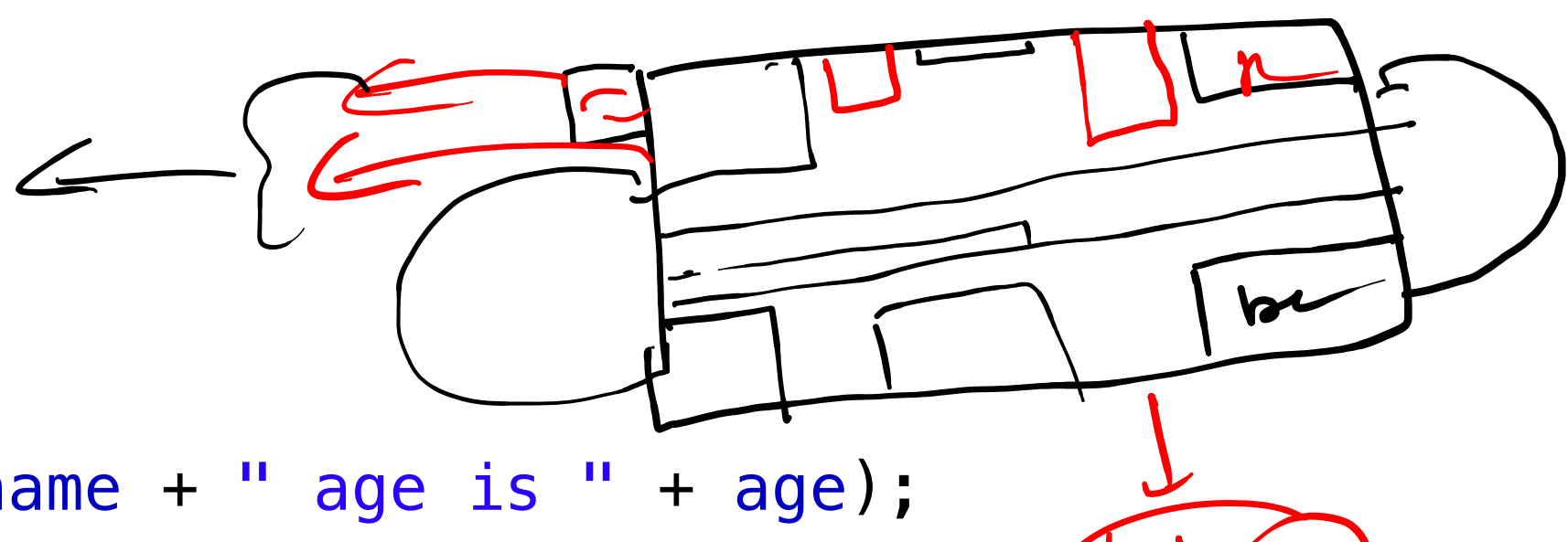


OOP

class

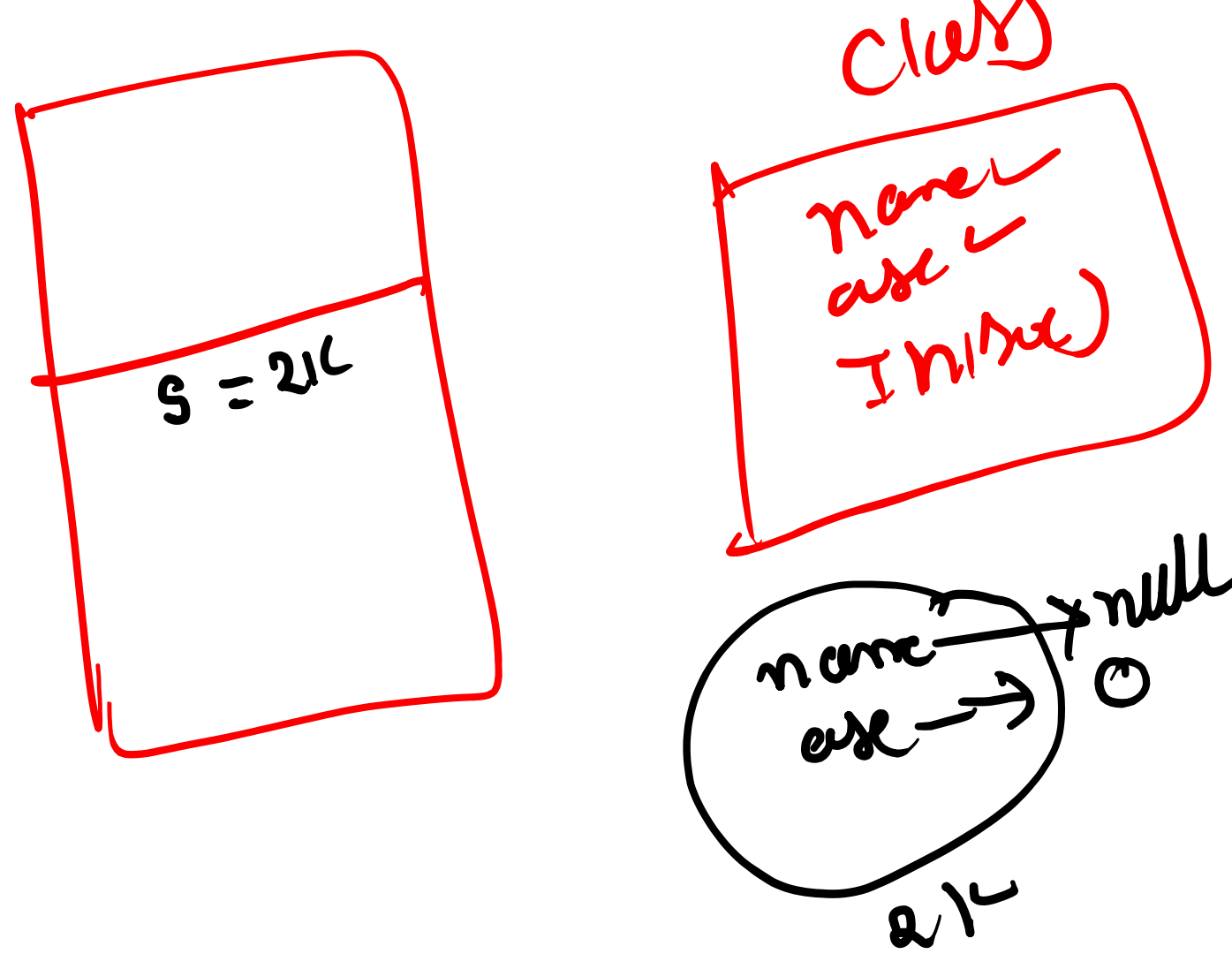


```
public class Student {
    String name;
    int age;
    public void Intro_yourSelf() {
        System.out.println("My name is " + name + " age is " + age);
    }
}
```



```
public class Student_Client {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Student s = new Student();
    }
}
```

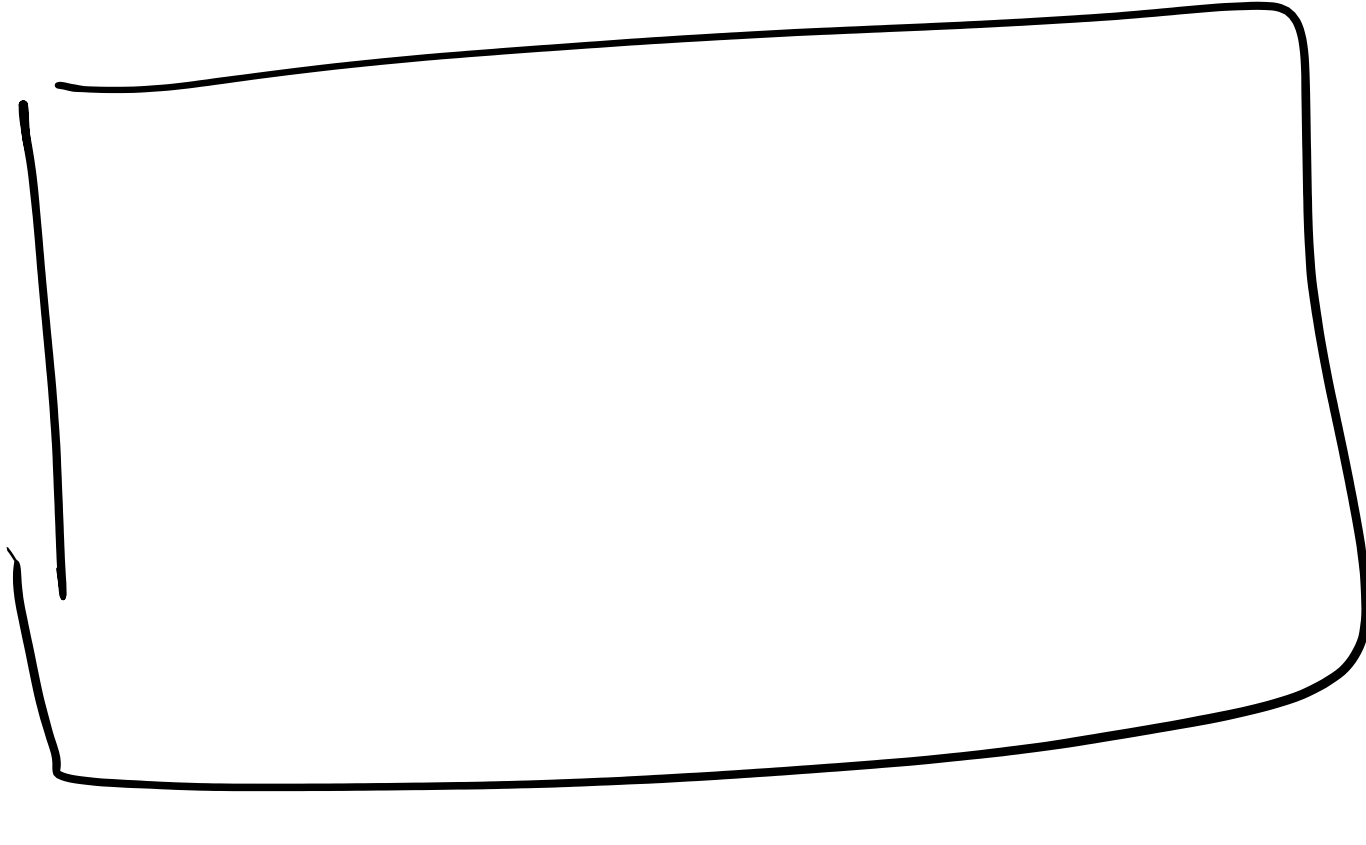
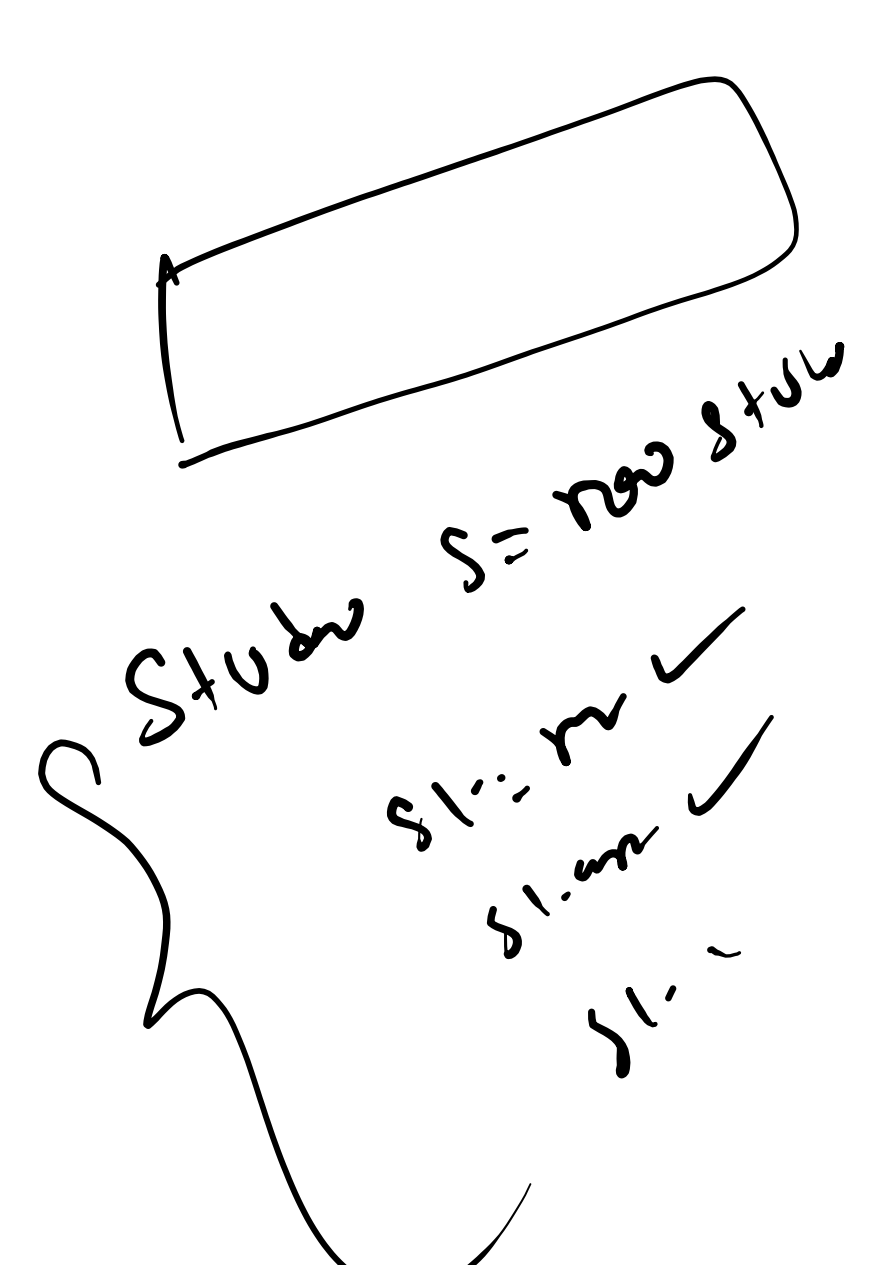
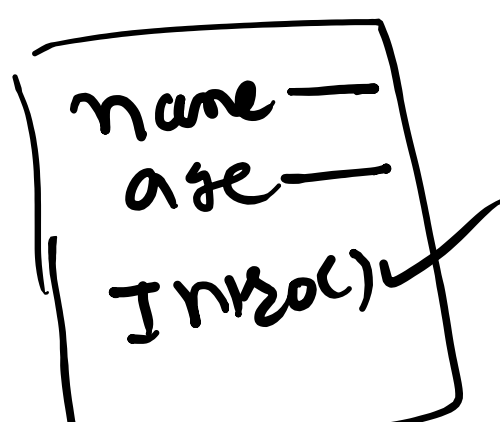
object



S1. Compare (S 2)

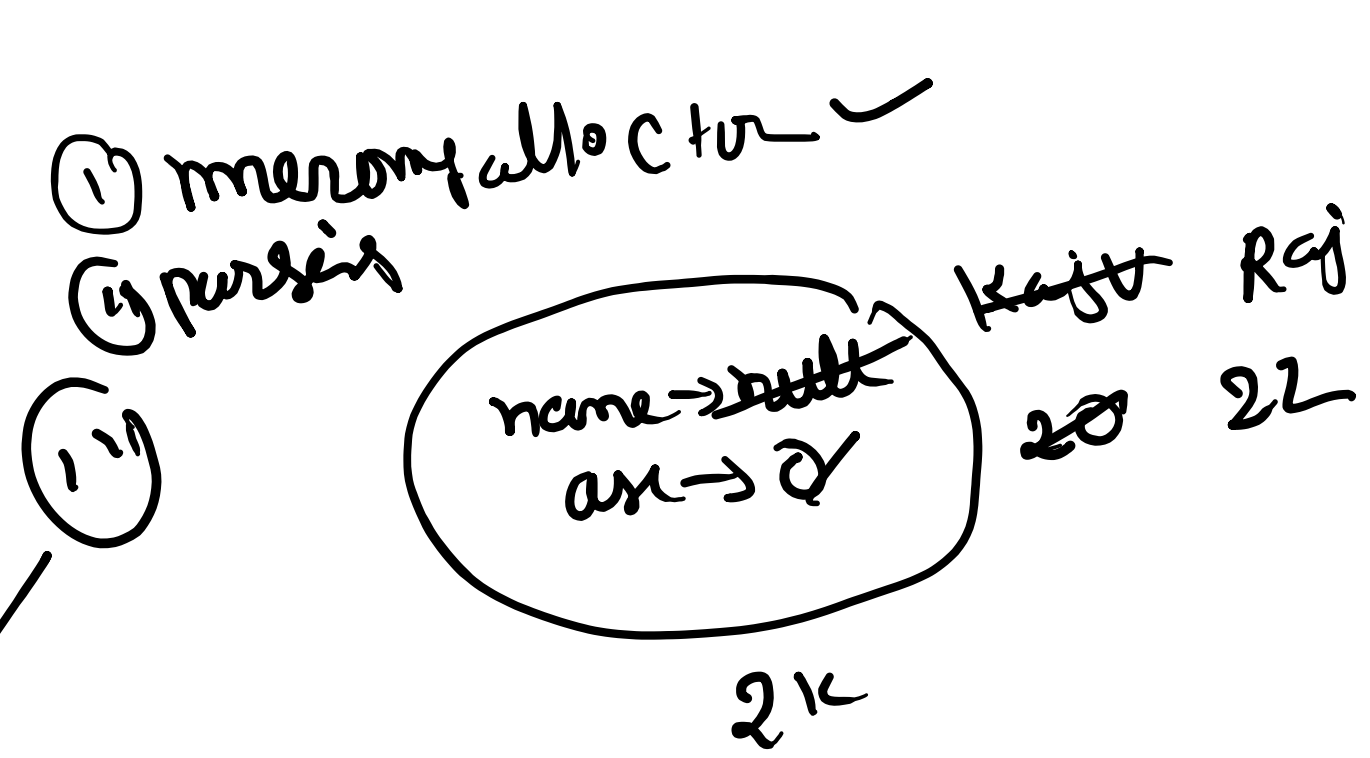
```
public class Student {
    String name;
    int age;
    public void Intro_yourSelf() {
        System.out.println("My name is " + name + " age is " + age);
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Student s1 = new Student();
        s1.name = "Kaju";
        s1.age = 19;
        s1.Intro_yourSelf();
        Student s2 = new Student();
        s2.name = "Raj";
        s2.age = 21;
        s2.Intro_yourSelf();
    }
}
```

this (20)



```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    Person p = new Person("Raj", 22);
    System.out.println(p.name);
    System.out.println(p.age);
    Person p1 = new Person();
    System.out.println(p1.name);
    System.out.println(p1.age);
}

public class Person {
    String name = "Kaju";
    int age = 20;
    public Person() {
    }
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
```



Public package

Encapsulation

- It is the process of **Grouping** the **data** with its related **methods** into single unit(Class).
- To **prevent unauthorized access** to data members of object.
- Advantages** of Encapsulation
 - Security
 - Code flexibility
 - Modularity
 - maintainability

Security

Error

Exception

