**Object-Oriented Programming (OOP) principles:**

1.**Class:** A blueprint or template.

Ex: Dog class defines what a dog is like (name, breed) and what it can do (bark).

2.**Object:** A specific instance of a class.

Ex: myDog is an object made from the Dog class with its own name and breed.

3.**Encapsulation:**

Encapsulation is a *process of wrapping code and data together into a single unit*, for example, a capsule which is mixed of several medicines.

We can create a fully encapsulated class in Java by making all the data members of the class private. Now we can use setter and getter methods to set and get the data in it.

Ex:

class Person {

public String getName() {

return name;

}

public void setN6ame(String name) {

this.name = name;

}

}

public class Main {

public static void main(String[] args) {

Person person = new Person();

person.setName("John");

System.out.println("Name: " + person.getName());

}

}

**4.Inheritance:**

Inheritance allows a new class (child or subclass) to inherit properties and methods from an existing class (parent or superclass).

Inheritance represents the IS-A relationship which is also known as a *parent-child* relationship.

**Types of Inheritance:**

**Single Inheritance:** A class inherits from one superclass.

**Multilevel Inheritance:** A class inherits from another class, and the child class is further inherited by another class.

Ex:

class Vehicle { }

class Car extends Vehicle { }

class SportsCar extends Car { }

**Hierarchical Inheritance:** Multiple classes inherit from a single superclass.

class Vehicle { }

class Car extends Vehicle { }

class Bike extends Vehicle { }

**Multiple Inheritance (via interfaces):** Java does not support multiple inheritance with classes directly to avoid complexity and ambiguity. However, it allows a class to implement multiple interfaces, which is a way to achieve multiple inheritance

**Hybrid Inheritance:** This is a combination of two or more types of inheritance.

**5.Polymorphism:**

It is a concept by which we can perform a single action in different ways.

Polymorphism can be achieved through method overloading and method overriding.

**6.Abstraction:** Abstraction is a process of hiding the implementation details and showing only functionality to the user. In Java, abstraction is achieved using abstract classes and interfaces.

**Super keyword:** Thesuper keyword  is a reference variable that is used to refer to parent class when we’re working with objects.

* Use of super with Variables, Methods, Constructors.

Ex:

class Parent {

int num = 10;

}

class Child extends Parent {

int num = 20;

void display() {

System.out.println("Child class num: " + num);

System.out.println("Parent class num: " + super.num);

}

}

public class Main {

public static void main(String[] args) {

Child obj = new Child();

obj.display();

}

}

**Super() Method:** super keyword to call a method from the superclass.

Ex:

class Parent {

void display() {

System.out.println("Parent class display method");

}

}

class Child extends Parent {

void display() {

super.display();

System.out.println("Child class display method");

}

}

public class Main {

public static void main(String[] args) {

Child obj = new Child();

obj.display();

}

}