**Day 4: 20-10-2025: OOPs concept**

**Polymorphism:** One name many forms or many implementation

2 types

1. Compile time polymorphism or static binding or early binding:

Method overloading: Method have same name and different parameter list (number of parameter list or type of parameter list must be different).

1. Run time polymorphism or dynamic binding or late binding:

Method Overriding: Method have same name and same method signature (number of parameter list, type of parameter list, and number of parameter list must be same with different logic inside a method).

**Annotation:** annotation is known as meta-data. Meta-data means data about data. Java provided lot of pre defined annotation. Few annotation we can use on class level or method level annotation or property/variable annotation.

Core java provided one the basic annotation ie @Override. This annotation we can use on method level. If method is overriding we doesn’t get any error else we get the error.

**abstract keyword**

1. abstract keyword we can use with method and class not with variable.
2. abstract method : the method without body or without burly braces or incomplete method is known as abstract method.

Syntax

abstract returnType methodName(parameterList);

abstract void speed();

1. abstract class : if class contains abstract method that class must be declare as abstract class.

abstract class Bike {

}

1. abstract class can contains normal as well as abstract method. means it can contain 1 or many or zero abstract method.
2. abstract class we can’t create the object. but we can create the reference. (memory can’t create using new keyboard).
3. class can be abstract but not mandatory it contains abstract method. use of this class(we can’t create the object of that class).
4. abstract class can contains default constructor. Even we can write empty as well as parameterized constructor.
5. Whichever class extends abstract class that class must be provide all abstract method belong to that class. that class can ignore only if that class itself is abstract class.