Day 17

Terms used in Inheritance:

- **Subclass / child class** → is a class which inherits other class.
  - also called derived / Extended class
- **Super class / Parent class** → a subclass inherit the features
  - also called Base class / Parent class
- **Extends** → Keyword used to inherit

POLYMORPHISM

- One thing is multiple forms / multiple ways

1. Method overloading (compile time P / early binding)
2. Method overriding (runtime P / late binding)

(2) METHOD overriding:

- Parent class & child class.
- method same but diff implementation
- Signature should be same

eg:

```
class A
{
    void M1()  ---> same method / signature
    {
        int a = 10;
        sysout(a);   } implementation
    }

class B extend A
{
    void M1()  ---> method same / signature
    {
        int a = 20;
        sysout(a);   } implementation.
    }
}

PSVM {
    // object create class B
    // b.M1();  ---> // 20
```

(IMP)

Parent class method child class overriding

if we want to access Parent details as well:

```
child class method
{
    super.phone();
}
```

super Keyword
↳ for accessing parent details
↳ use in child class method

Automatically do:

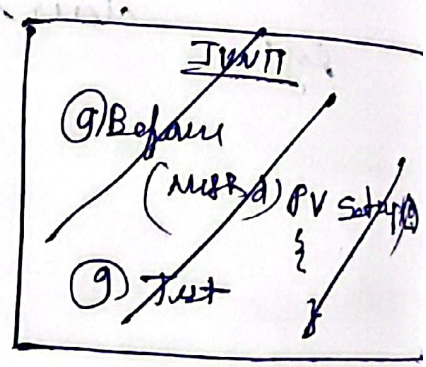
Right click → source → override / implement methods

→ can.

@override (Annotations):

→ we can't change structure after this

@Before
```
public void setup()
{
    // define activity
}
```



@Test
```
Public void test()
{
    // test activity
}
```

Why Runtime (P) / late binding?

Signature check at Execution time.

(2) Method overloading:

- Simple class multiple times same name method create

eg: class A {

add ()

add (int a) add (int a, int b)

add (double a, double b)

}

Rules:

(1) would by changing no of arguments / parameters

(2) would by changing the data types of arguments

(3) would by changing the order of Parameters

↳ //add (int a, double b)

//add (double a, int b)

→ How method name is same But Arguments are getting changed

Compile time Polymorphism:

(1) whenever an object is bound with this functionality at the compile time it is known as Compile time (P)

(2) At compile time Java knows which method to call by checking the method signatures.

(3) Method overloading says you can have more than one function with same name in one class having a diff Prototype

(4) Method can be overloading if and only if method has the same name but with diff signature.

⑤ Signature Mean no of arguments, type of argument & order of arguments : public void sum()
{
int a=10;
sysout(a);
}

- There are 3 ways to overload method:

① by changing no of argument / Parameter
② by changing the datatypes of arguments
③ by changing the orders of Parameters.

② METHOD OVERRIDING:

① Declaring a method in subclass which is already present in Parent class
② Overriding is done so that a child class can view its own implementation to a method which is already provided by the Parent class.
③ Method should share same Signature in Method overriding

Super Keyword:

- super keyword in java is a Refrence variable which is used to refer immediate Parent class object

Usage:

① Super can be used to refer immediate Parent class instance variable.
② Super can be used to invoke immediate Parent class method
③ Super can be used to invoke immediate Parent class constructor