Polymorphism

Many behaviour

Compile time

Early binding

↓

over loading

↓

add (int, int),
add (int, int, int)

which method to take decided at compile time.

Run time

late binding

↓

(overridding)

A add(int, int)

↑

B add (int, int)

decided at runtim

→ To implement run time polymorphism, we do with the help of

Dynamic Method dispatch

```
Clas A{

      void show( )   { sout(" in A show"); }
}

Clas   B extends A {
      void show()   { sout ("in B show"); }
  }

Clas C   extends A {


  }

Puplic clas Demo{

      psvm (      ){

            B obj = new B();

            obj.show();
```

Can we write,

A obj = new B();

Note

the obj type is A but the implementation is B.

So, bcoz we can refer type of obj as parent but the object can be of child.

Yeah, it is possible to create a ~~reference~~ reference of superclass ~~a~~ but the object of subclass

→ Now, copy same method in class B.

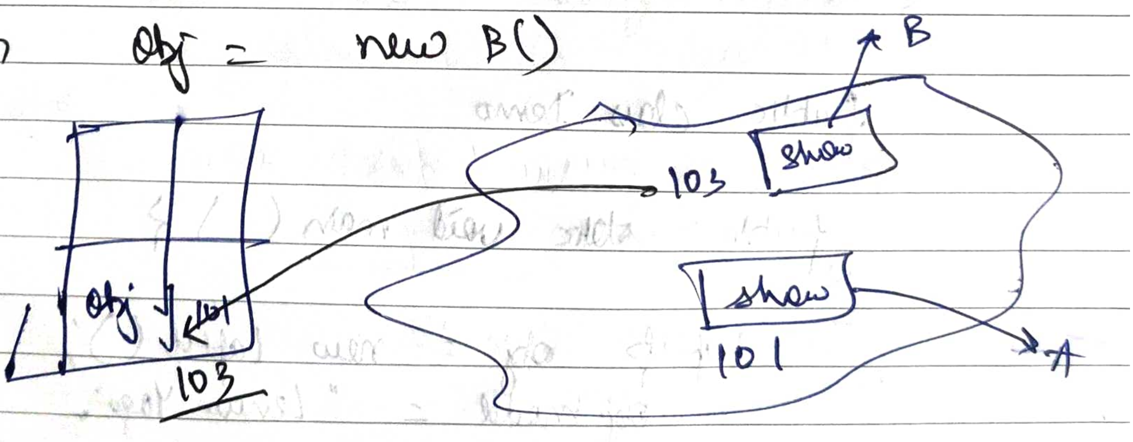
~~A a~~ so overridding occurs.

```
A obj = new B();
obj.show()
   ↓
in B show.
```

```
A obj = new A()
obj.show()
   ↓
in A show.
```

```
Now   obj = new B()
```

i.e.



So, ~~typ~~ type is A, but obj is B

```
So, obj.show();
       ↓
in B show.
```

```
obj = new C().
obj.show() → in C show.
```

Note

Irrespective of the type of object, it depends on what objects we have

(Only Possible in Inheritance)