Differences Letwern Mothol overloading & Mothod overriging

METHOD OVER LOADING METHOD DVERRIDING

the apply that many a now

Re-veing same name for Super 4 sub-class methods with different number of arguments (or) with different type of arguments with sor; without same return type within a class is called nother overloading

- > No of arguments type of arguments must be different
- -> It doesn't signify returns, type.
- -> No method hills mo another method
- Inheritance is not involved
- -> It is a compile time / states
 Polymorphism

Super & Sub - class methods
with same number of
arguments, with same type
of arguments & with same
return type is called
Method overriding

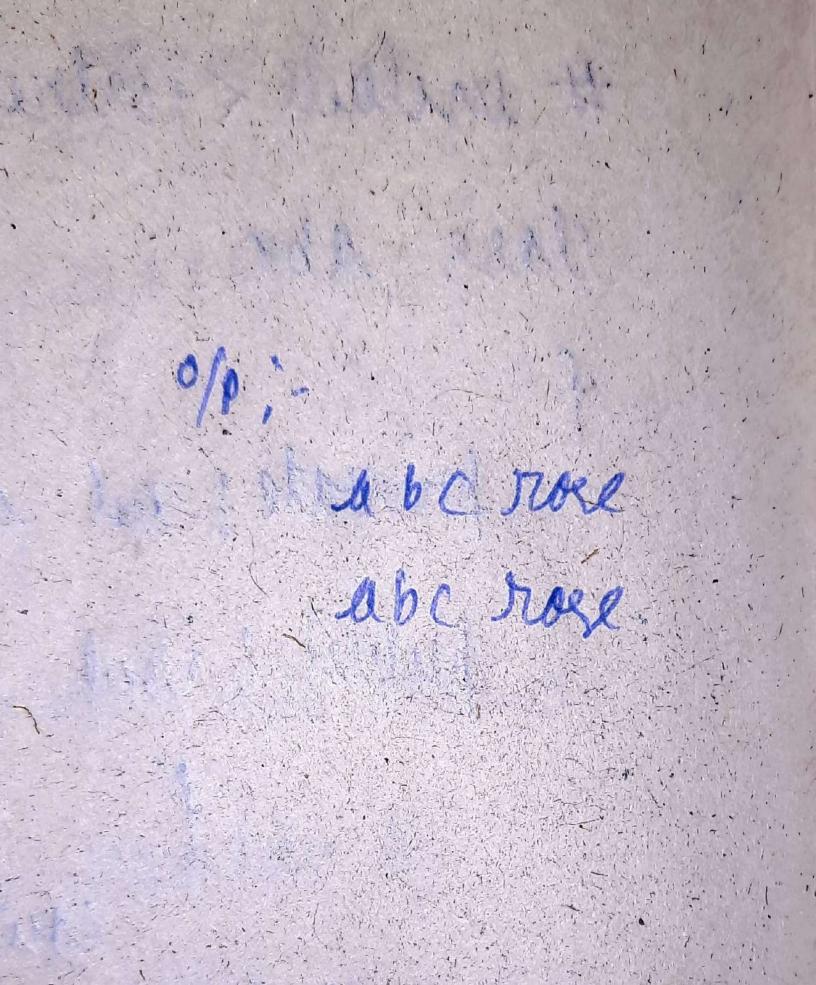
-> No of orguments 1 type of arguments must be some

Comment

- -> et signifies returne)
- -> Sub class method hides Superclass method
- -> unheritance is involved
- -> It leads to Dynamic binding

```
Constantial and
-> wing namespace std;
  # include < i ostream >
                                  BINE SHALLEN SOUTH
                                   a cee . They
  class Abc
  2 private: int a, b;
    public : void road ()
             coul << "enter a, b";
             um >> a >> b;
           void write ()
            cont « a « « "
  class xyz: public a bc
   private: int n, y;
     public: void read ()
             cont < c " enter x, y"
             cin >> x >> y;
           void write ()
            cout << x << "
   main()
    e. real (); p. write (); p. abc: real(); p. abc: write ();
```

```
the marking and
COMPILE TIME BINDING! -
                               The sand the sand of the
-> Using namespace stil;
                              # include < i ortran >
 class abc
                               I MADORE
                              abc rose
    fullie: voiil rose ()
             cout << " a bc rose"
                     THE COLLECTION OF ME
  class xy 2: pullir abc
                            The strains
   public: void sure ()
                            o Street Co
           cout << "xy & rose";
                          "是我们是这是我们
                          Legit True thank
  main ()
                           the extensi
      abc *p,91;
      xy2 9;
      P=49;
      p -> Just ()
                        and the second
      \rho = 4\pi
      p >> rose ();
```



If we run above program abouters is called, because the pointer is abouters the building is compute Timo duriding.

without () postpondes the decision at dunding at compile time of decision is taken at Run-time of the program.

VIRTUALL) SYNTAX :-

reintered name (....)

body 3

RUNTIME BINDING :-

-> using namerhaie sid; # include < i ortream>

class abc

fulli: rivilual void rose ()

¿ cout « "abc rose";

class xyz: public abc

t judic! void rose ()

Coud << " x y & rose";

main () without the product in the same of the same abc *p,n; 0/0: xyz rose P= Lay was the world of the last of the la Ab c ruse MANAGE P= kn: P-> rose ();