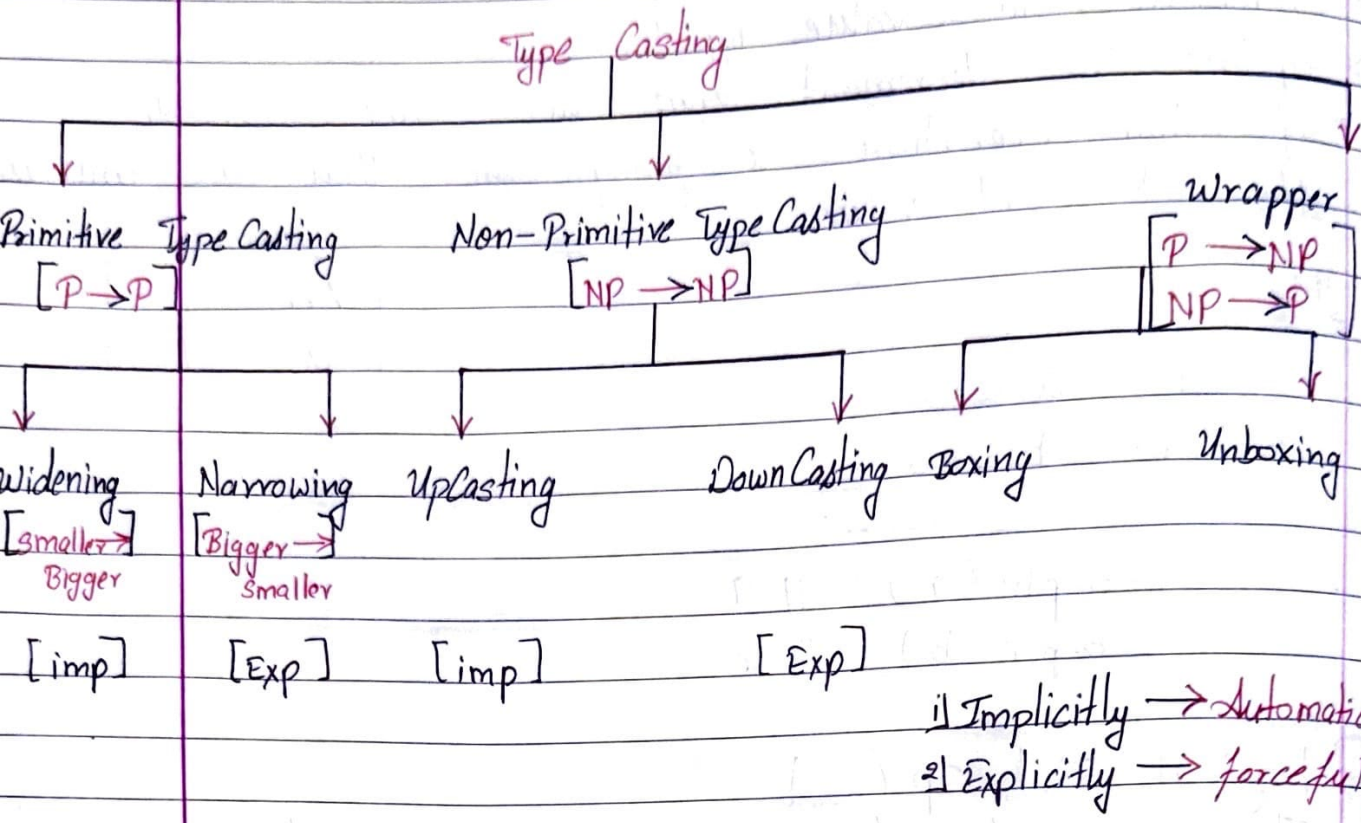
Type Casting



1] Implicitly → Automatically
2] Explicitly → forcefully

i] Type Casting

→ In this process we convert one datatype to another datatype.

→ we have few types of Type Casting.

1] Primitive Type Casting
2] Non-Primitive Type Casting
3] Non-Primitive to Primitive [NP → P] or Primitive to Non-Primitive [P → NP]

① Primitive Type Casting

→ The Process of converting one Primitive datatype into another Primitive datatype is known as P.T.C.

#) In this we have 2-Types.

i) Widening
ii) Narrowing

i) Widening

→ In Widening we convert smaller Primitive DT into Bigger Primitive DT.

→ In Widening there is No Data loss, Hence it is also known as Implicit Type Casting.

→ Widening is done by the Compiler.

Eg:→ i)

```
Class P1
{
    public static void main(String[] args)
    {
        byte b = 12;
        int a = b;
        System.out.println(a);
    }
}
```

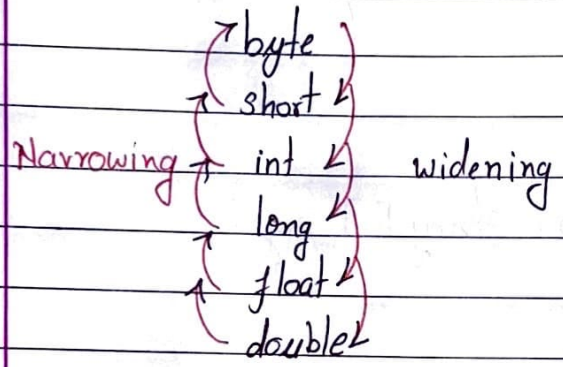
O/P:→ 12

* Narrowing
→ In Narrowing we convert Bigger Primitive DT into smaller Primitive DT.
→ In Narrowing we have Data loss, Hence it is also known as Explicit Type Casting.
→ Narrowing is done by the Programmer.
→ Narrowing is done with the help of TypeCast operator.

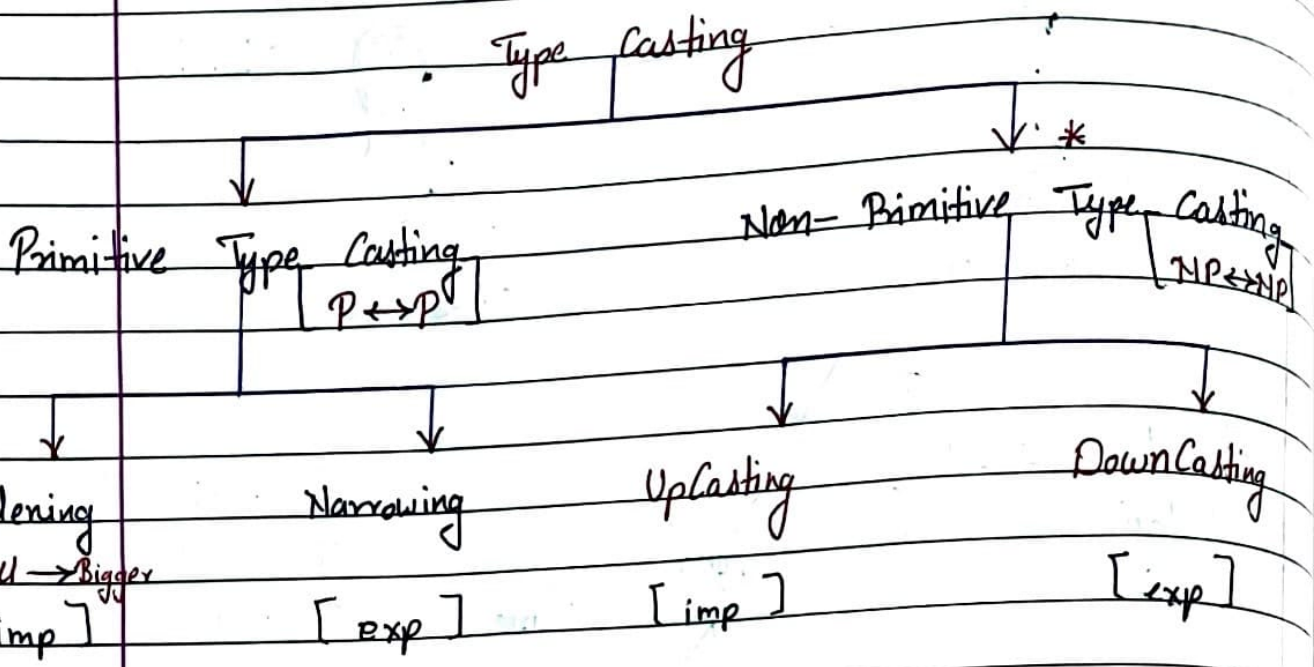
datatype var = (Converting DT). Converting VarIVal.

byte b = (byte) a;

* Type Casting Chart



Type Casting



#) Non-Primitive Type Casting [N.P.T.C]

→ The process of converting one Non-primitive data type into another Non-primitive data type is known as N.P.T.C.

i) UpCasting

→ The process of converting sub-class type into super-class type is known as UpCasting.

Note :→

i) UpCasting will be done implicitly by the Compiler.

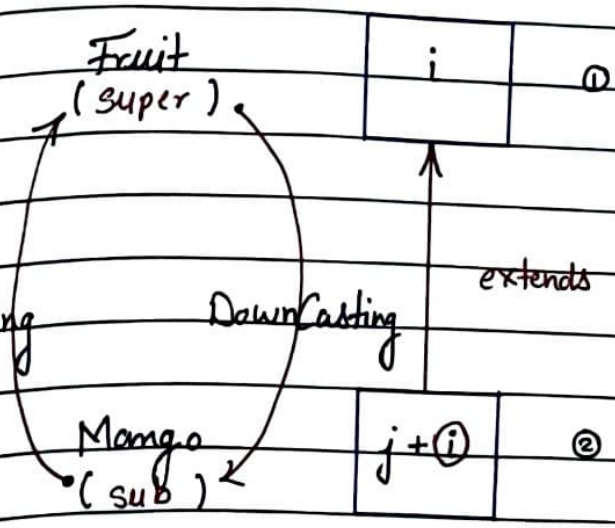
2) DownCasting

→ The process of converting super-class type into sub-class type is known as DownCasting.

Note :→

2) DownCasting will be done explicitly by the programmer.

*



i) Fruit f = new Fruit();
(super) (super)

2) Mango m = new Mango();
(sub) (sub)

3) Fruit f = new Mango();
(super) (sub)

UpCasting

Note :->

i) with the help of sub-class object reference we can use members of sub-class as well as members of super-class

2) with the help of super-class object reference we can use only the members of super-class but we can not use members of sub-class.