

Q:- Distinguish Widening & Narrowing with Example.

A:- Type Casting :- type casting is a method or process that converts a data type into another data type in both ways manually and automatically.

- ⇒ Automatic conversion is done by the compiler.
- ⇒ Manual conversion performed by the programmer.

Type casting is divided into 2 types.

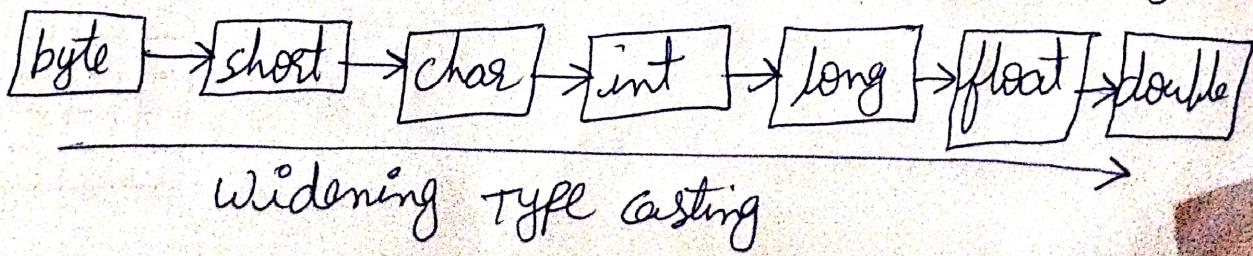
1. Widening type casting
2. Narrowing type casting

1. Widening Type Casting :- Converting a lower data type into a higher one is called Widening type casting.

also known as Implicit conversion or Casting down. it is done automatically.

⇒ it is safe because there is no chance to lose data. it takes place when;

1. both data types must be compatible with each other.
2. The target must be larger than the source type.



→ the conversion between numeric data types to char or Boolean is not done automatically.

Ex:-

class widening

{

    public static void main(String[] args)

{

        int x=7;

        long y=x;

        float z=y;

    System.out.println("Before conversion, int value "+x);

    System.out.println("After conversion, long value "+y);

    System.out.println("After conversion, float value "+z);

O/P:-

Before conversion : 7

After conversion, long value is : 7

After conversion, float value is : 7.0

Q. Narrowing type casting :-

→ it convert a higher data type into a lower one.

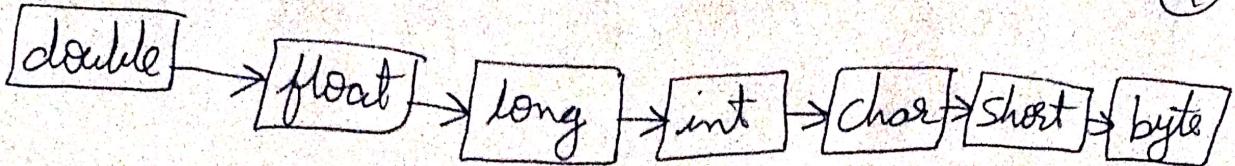
→ it also called as Explicit or Casting up conversion.

→ if it is done manually by the programmer.

→ if we not perform casting then the compiler reports a compile-time error.



(2)



Narrowing Type Casting.

1. we have converted double type into long data type.
2. then long data type is converted into int type.

Ex:-

class Narrowing

{ public static void main (String args[]) }

    double d = 166.66;

    long l = (long)d;

    int i = (int)l;

System.out.println("Before conversion: "+d);

System.out.println("After conversion into long type "+l);

System.out.println("After conversion into int type "+i);

} } O/P:- Before conversion: 166.66

After conversion into long : 166

After conversion into int : 166



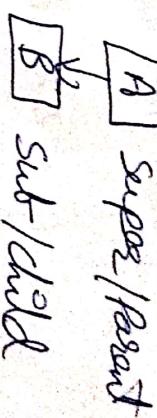
### 30:- Uses of SUPER & Super class reference to

Sub-class object?

- i) SUPER keyword - Super keyword is a reference variable which is used to refer parent class object.

⇒ whenever you create the instance of sub-class, an instance of Parent class is created implicitly by super reference variable.

Usage of SUPER keyword :-



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.

```
Ex:-
    class Animal {
        void eat() {
            System.out.println("eating..");
        }
    }
```

```
class Dog extends Animal {
```

```
    void eat() { S.O.PNL("eating bread"); }
```

```
    void bark() { S.O.PNL("barking"); }
```

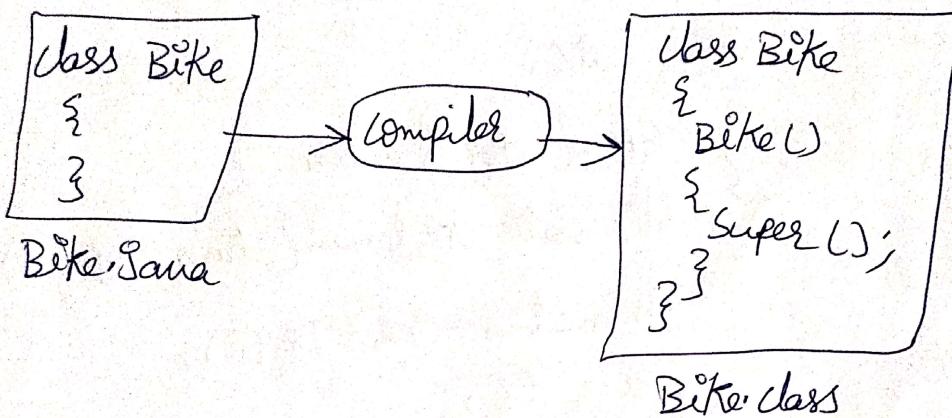
```
    void work() { super.eat(); }
```

```
    void bark(); }
```

(4) 15  
(3)

```
class TestSuper2 {  
    public static void main(String args[]) {  
        Dog d = new Dog();  
        d.work();  
    }  
    op:- eating..  
         barking..
```

Note:- super() is added in each class constructor automatically by compiler if there is no super() or this().



Q:- Static Binding & Dynamic Binding with Examp.

A:- Binding :- Connecting a method call to method Body is called Binding.

→ 2 types of Binding

1. Static (Early) Binding.

2. Dynamic (Late) Binding.

Error:- it thrown when JVM is out of resources.

Static Binding :- when type of the object is determined at compile time (by compiler) is called static binding.

⇒ if there is any private, final or static method in a class there is static binding.

Ex:- Dynamic Binding :- when the type of object is determined at Run-time is called dynamic binding.

class Animal

{ void eat()

{ System.out.println("animal is eating..");

class Dog extends Animal

{ void eat()

{ System.out.println("dog is eating..");

public static void main(String args[])

{ Animal a = new Dog();

} a.eat();

OP:- dog is eating.

(4)

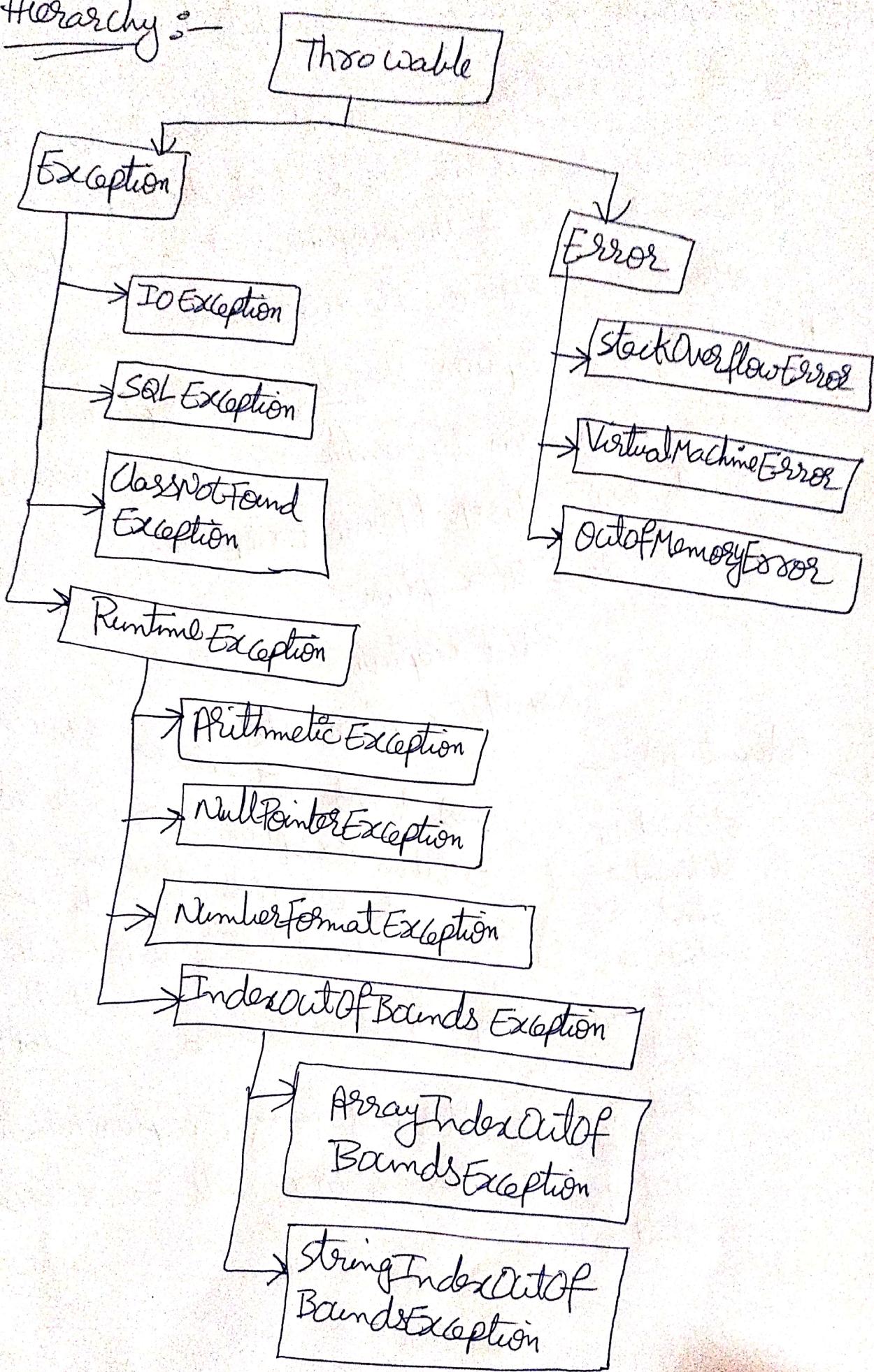
## 5Q:- Exception Hierarchy of Benefits :

Ques:- what is Exception :- means Abnormal condition -  
 → an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at run-time.

Ques:- what is Exception Handling :- Exception Handling is a mechanism to handle run-time errors such as:-  
 1. ClassNotFoundException.  
 2. IOException.  
 3. SQLException.  
 4. RemoteException etc.

Advantages :-  
 Statement 1;  
 Statement 2; } rest of code not executed  
 Stmt 3; } when Exception/error occurred  
 Stmt 4; } at 5th line, program runs  
 Stmt 5; //exception (error) occurred. to 5th line and stop because  
 Stmt 6; } of error 6 to 10  
 Stmt 7; } Stmts are stopped.  
 Stmt 8; }  
 Stmt 9; } after performing Exception handling  
 Stmt 10. } then 6 to 10 will run successfully.

Hierarchy :-



1. IOException:- IOExceptions are Input/output exceptions, they occur whenever an input or output operation is failed or interpreted. Ex:- trying to read a file that not exist, it throws at compile-time.
2. SQLException:- it provide information about errors and warnings that occur while a datasource is being accessed. It consists of ~~of~~ JDBC, database errors.
3. ClassNotFoundException:- it occurs when the application tries to load a class but classloader is not able to find in classpath.
4. Runtime Exception:- it is the superclass of all classes that are thrown during normal operation of JVM (Java Virtual Machine).
5. ArithmaticException:- it is most powerful mechanism to handle the runtime errors. If something is divided by zero (5/0) it shows ArithmaticException.
6. NullPointerException:- whenever one tries to use a reference that has the null value, it shows NullPointerException.
7. NumberFormatException:- it is thrown when we try to convert a string into a numeric <sup>value</sup> float or int.

8. IndexOutOfBoundsException:- it is thrown when you access the element at an index of a type (String, array etc) beyond its range.

Array size (2)

-1 or +3 then it shows error.

9. ArrayListOutOfBoundsException:- it is an object that stores a fixed-size sequential collection elements of same type.

10. StringIndexOutOfBoundsException:- it is thrown when you access string at index which is greater than its length.

### Errors :-

1. StackOverflowError:- it occurs when we haven't provided the proper termination condition to our recursive function, which means it will turn into an infinite loop.

2. VirtualMachineError:- it thrown when JVM is broken or run out of resources.

3. OutOfMemoryError:- it thrown when there is insufficient space to allocate an object in Java heap.