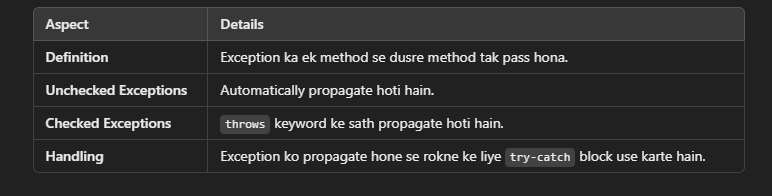
* *checked exception me autoducking nahi hoti hai lekin hum Throws keyword se throw kar sakte hai exception unchecked exception me autoducking / automatic mebhi aisa hi kuch hota hai internaly*
* *unchecked* exception me autoducking automatic hoti hai

#Throws

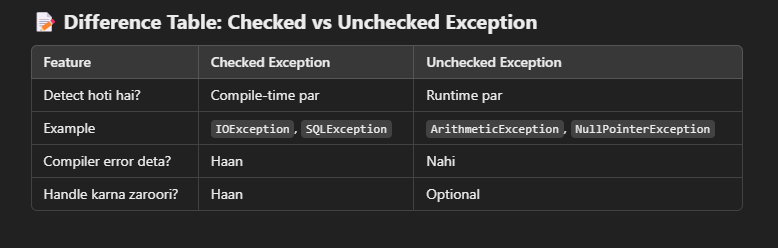
* It is required only for checked exception and for unchecked exception there is no use

***Deffi*** *.* we can use to delegate responsibility of exception handling to the caller (jvm or onother methods)

# Exception propagation



# Re Throwing Exception

* Manlo 10/0 ab arithemethic exception aayegi or use apn ne try-catch block se handel kar liya hai or catch ke andar apn throw keyword se apn nullpointer exeption generate kar raha hai matlab ek exception ko dusri exeption me convert kar rahe hai **Checked Exception**: Compiler ko pata hoti hai, handle karna zaroori hai.
* **Unchecked Exception**: Runtime par aati hai, handle karna zaroori nahi hai

# To String Method

* Override Method

 public String toString() {

        return "Person [id=" + id + ", name=" + name + ", salary=" + salary + "]";

    }

Agar ham Bina to string method override kiye bina agar ham object ka reffence ko print karte hai ya use karte hai or tostring method override nahi hai to baha Obejct class ki tostring method ke pass jayega or object class ki method ka default bihebiyar means object ka address print means address return karna hota hai

**Example** :

        Person p1 = new Person(101, "deepak", 10000);

        System.out.println(p);

# Equals Method

* Override Method
* public boolean equals(Person2 p) {
* return this.id == p.id && this.name.equals(p.name) && this.salary == p.salary;
* }

Agar ham equals method ko override nahi karte hai or ham do object ko equals methods se compare karte hai to to baha Object class ki default bihebiyar bali equals methods ko call karta hai or uska default behibiyar object ka address ko return karta hai or compare address karta hai or value compare nahi karta agar aap chahte hai ki object ki value ko compare kare to apko equals method ko override karna padega ^

**Example** :

        Person2 p = new Person2(101, "deepak", 10000);

        Person2 p1 = new Person2(101, "deepak", 10000);

        System.out.println(p.equals(p1));

# HashCode Method

public int hashCode() {

        return Objects.hash(id, name, salary);

    }

Hashcode method Agar apne override nahi kiya or aap do object ke reference ka hashcode nikal rahe hai or value same hai fir bhi hashcode deferense print karega kyuki baha Object class ki hashcode method call karega jiska default bihebiyar method memory address ke basis par hashcode generate karna hai agar aap is tarah se method ko override karte hai to baha “id,name,salary” ka hashcode generate karega joki dono me same hai fir baha same hashcode print karega

**Example** :

        Person3 p = new Person3(101, "deepak", 10000);

        Person3 p1 = new Person3(101, "deepak", 10000);

        System.out.println(p.hashCode());

        System.out.println(p1.hashCode());