

LAB PROGRAM VII

Q Write a programme that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extend the base class. In father class, implement a constructor which takes the age and throws exception WrongAge() when the input age  $\leq 0$ . In son class implement a constructor that uses father and son's age and throws an exception if son's age  $\geq$  father's age.

```
class WrongAge extends Exception {  
    public WrongAge (String message)  
    {  
        super (message);  
    }  
}
```

```
class Father {  
    int fatherAge;  
    public Father (int age) throws WrongAge {  
        if (age  $\leq 0$ ) {  
            throw new WrongAge ("Father's age can't be  
negative");  
        }  
        this.fatherAge = age;  
    }  
}
```

```
class Son extends Father {
    int sonAge;
    public Son (int fatherAge, int sonAge) throws
        WrongAge {
        super (fatherAge);
        if (sonAge < 0) {
            throw new WrongAge ("son's age can't be
            negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge ("son's age can't be
            greater (or equals father's age)");
        }
        this.sonAge = sonAge;
    }
}

public class Main {
    public static void main (String[] args) {
        try {
            Son son3 = new Son (-1, 10);
            Son son1 = new Son (25, 50);
            System.out.println ("Father's Age: " + son
            fatherAge + ", Son's Age: " + son1.sonAge);
        }
        catch (WrongAge e) {
            System.out.println ("Exception caught: " + e.get
            message());
        }
    }
}
```



Output:

~~Enter internal marks (<=30):~~

~~Son Age can't be negative~~

~~Son Age can't be greater than father's Age~~

~~Father's Age: 50, Son's Age: 25~~

Rs

21/12/24

(50,25);

# lab 7

```
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class Father {
    int age;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Father's age cannot be negative.");
        }
        this.age = age;
    }
}

class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equ
        }
        this.sonAge = sonAge;
    }
}

public class InheritanceExceptionDemo {
    public static void main(String[] args) {
        try {
            Father father = new Father(40);
            Son son = new Son(40, 20);
            System.out.println("Father's age: " + father.age);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }

        try {
```

```
        Father father = new Father(-5);
    } catch (WrongAgeException e) {
        System.out.println("Exception: " + e.getMessage());
    }

    try {
        Son son = new Son(30, 30);
    } catch (WrongAgeException e) {
        System.out.println("Exception: " + e.getMessage());
    }
}
}
```

## ▼ Output

```
Father's age: 40
Son's age: 20
Exception: Father's age cannot be negative.
Exception: Son's age cannot be greater than or equal to Father's age.
```