

16/2/24

LAB-7

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

Code:

```

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

class Father {
    int Age;

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

    public int getAge() {
        return age;
    }
}

class Son extends Father {
    int SonAge;

    public Son(int fatherAge, int SonAge) throws WrongAge {
        super(fatherAge);
        if (SonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's Age");
        }
        this.SonAge = SonAge;
    }
}

```



```
public int getSonAge() {  
    return sonAge;  
}
```

```
}  
  
public class ExceptionInheritanceDemo {  
    public static void main (String [] args) {  
        try {  
            Father father = new Father (30);  
            System.out.println ("Father's Age: " +  
                                father.getAge());  
            Son son = new Son (30, 20);  
            System.out.println ("Son's age: " +  
                                son.getSonAge());  
        }  
        catch (WrongAge e) {  
            System.out.println ("Exception caught: "  
                                + e.getMessage());  
        }  
    }  
}
```

### Algorithm :-

- Step 1: Start
- Step 2: Create class WrongAge extending Exception class
- Step 3: Creating constructor of WrongAge (String message)  
passing message to constructor of super class.
- Step 4: Create a base class Father
- Step 5: Declare class variable int Age
- Step 6: Create class constructor passing int Age, throws WrongAge  
and checking if age < 0  
throw new WrongAge Exception  
having message ("Age cannot be Negative");
- Step 7: Create a method int getAge() returning Age value;



Step 8: Create a Derived class Son extending Father class

Step 9: In Son class constructor, it takes both FatherAge and SonAge and throws WrongAge Exception if  $\text{SonAge} > \text{FatherAge}$ , printing message "Son Age cannot be greater than Father Age"

Step 10: Create a method int getSonAge() to return SonAge value

Step 11: Create Main class as ExceptionInheritanceDemo and main method

Step 12: Inside main method, in try block create object of class Father as father having age 30.

Step 13: Create object of Son class as son and passing (30, 20) as FatherAge and SonAge

Step 14: Print Father Age and Son Age

Step 15: Catch wrong age error in try block and print the resultant getMessage value.

Step 16: Stop

Output:

1> Enter Father's Age : -5

WrongAgeException Caught : Age Cannot Be Negative

2> Enter Father's Age : 40

Enter Son's : 50

WrongAgeException Caught : Son Age Cannot Be greater than Father Age

3> Enter Father's Age : 40

Enter Son's Age : 20

Father's Age : 40

Son's Age : 20