

PROGRAM 7:

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 uses both father and son’s age and throws an exception if son’s age is >=father’s age.

CODE:

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

class Father {
    int fatherAge;

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

```

    }

    public int getFatherAge() {
        return this.fatherAge;
    }
}

class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        // Call the constructor of the Father class
        super(fatherAge);

        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be negative.");
        }
        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 this.sonAge;
    }
}

public class EX {
    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.getFatherAge());
        System.out.println("Son's age: " + son.getSonAge());

        //    Son invalidSon = new Son(40, 45); // Throws exception
        //    Son invalidAgeSon = new Son(40, -5); // Throws exception

    } catch (WrongAge e) {
        System.out.println("Exception: " + e.getMessage());
    }
}
}
```

OUTPUT:

```
Father's age: 40
Son's age: 20
```

NOTES:

Lab-7.

PAGE NO.:

DATE:

1. write a program that demonstrates handling of exceptions in inheritance tree. create a base class called "Father" and derive of class called "son" which extends the base class. In Father class implement a constructor which takes the age and throws the exception wrong age() when input age < 0. the son class, implement a constructor that uses both father and son's age and throws an exception if son's age is > = 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 < 0) {  
            throw new wrongage ("Father's age  
            cannot be negative");  
        }  
        this.fatherage = age;  
    }  
}
```

```
class son extends Father {  
    int sonage;  
    public son (int fatherage, int sonage) {
```

```

super (fatherage);
if (sonage >= fatherage) {
    throw new wrongage ("son's age
    cannot be greater or equal father age");
}

```

```

this.son = sonage;
}
}

```

```

public class main {

```

```

    psvm (?)

```

```

    try {

```

```

        // Son son3 = new Son (-1, 10);

```

```

        Son son1 = new Son (50, 25);

```

```

        sout ("father age " + son1.fatherage
        + "son's age" + son1.sonage);

```

```

    }

```

```

    catch (wrong age e) {

```

```

        sout ("Exception caught " + e.

```

```

        getMessage());

```

```

    }

```

```

}

```

Output

Father's age. 50

Son's age. 25

12