Lab 7 WAP that demonstrates handling of exceptions in inheritance tree. Create a base class "Father" and a derived class "Son". 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 son's age and throws an exception if son's age is father's age.

```
import java.util.*;
class WrongAge extends Exception {
    WrongAge(String s) {
        super(s);
    }
}

class InputScanner {
    Scanner s = new Scanner(System.in);
}

class Father extends InputScanner {
    int fatherAge;
    Father() throws WrongAge {
        System.out.println("Enter father's age");
        fatherAge = s.nextInt();
        if (fatherAge < 0)
            throw new WrongAge("Age
            cannot be negative");
    }
    void display() {
        System.out.println("\n Father's age:"
            + fatherAge);
    }
}
```

```
class Son extends Father {
    int sonAge;
    Son() throws WrongAge {
        super();
        System.out.println("Enter son's age:");
        sonAge = s.nextInt();
        if (sonAge < 0)
            throw new WrongAge("Age cannot be negative.");
        else if (sonAge >= fatherAge)
            throw new WrongAge("Son's age cannot be greater than father");
        else {
            displayf();
            displayson();
            throw new WrongAge("Valid Age");
        }
    }
}
    void displayson() {
        System.out.println("\n Son's age " + sonAge);
    }
}

class Main {
    public static void main(String[] args) {
        try {
            Son son = new Son();
        } catch (WrongAge e) {
            System.err.println(e.getMessage());
        }
    }
}
```

Output:

```
Enter father's age
43
Enter son's age
23
Father's age : 43
Son's age : 23
Valid age

Enter father's age
-2
Age cannot be negative

Enter father's age
43
Enter son's age
46
Son's age cannot be greater than father's age
```