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=father’s age.

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

class Father {
    int fatherAge;

    Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Father's age cannot be negative!");
        }
        this.fatherAge = age;
        System.out.println("Father's age: " + fatherAge);
    }
}

class Son extends Father {
```

```
    int sonAge;

    Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age must be less than Father's age!");
        }

        this.sonAge = sonAge;
        System.out.println("Son's age: " + sonAge);
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            Son s1 = new Son(45, 20);
            System.out.println("Object created successfully!\n");

            Son s2 = new Son(40, 40);

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

```
        try {
            Son s3 = new Son(-50, 10);


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

OUTPUT: