

Write a Program that demonstrate handling of Exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son", which extends the base class, implement a constructor which takes the age and throws the exception Wrong Age, when the input age < 0. In Son class implement a constructor that takes both Father & Son's age & throws an exception if Son's age is > = father's age.

```
import java.util.Scanner;

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

class Father {
    protected int FatherAge;

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

class Son extends Father {
    private int SonAge;

    public Son (int FatherAge, int SonAge) throws WrongAge {
        super (FatherAge);
    }
}
```

```

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

```

```

if (SonAge >= fatherAge) {
    throw new WrongAge ("Son's age cannot be greater than or equal to father's age");
}
}

```

```

public class Main25 {
    public static void main (String[] args) {
        Scanner scanner = new Scanner (System.in);
        try {
            System.out.print ("Enter father's age:");
            int fatherAge = scanner.nextInt();
            System.out.print ("Enter son's age:");
            int sonAge = scanner.nextInt();
            Son son = new Son (fatherAge, sonAge);
            System.out.println ("Father's age: " + fatherAge);
            System.out.println ("Son's age: " + sonAge);
        } catch (WrongAge e) {
            System.out.println ("Exception Caught: " + e);
            System.out.println ("Exception Caught: " + e.getMessage());
        }
    }
}

```

```

Catch (Exception e) {
    System.out.println ("Error:" + e);
    System.out.println ("Error:" + e.getMessage());
}
finally {
    Scanner.close();
}
}
}
}

```

Output:  
Enter father's age: 50  
Enter son's age: -10

Exception Caught: Wrong Age: Son's age Cannot be  
negative @ Zero

Enter father's age: 30  
Enter son's age: 40

Exception Caught: Wrong Age: Son's age Cannot  
be greater than or Equal to  
father's age

```
import java.util.Scanner;

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

class Father {
    protected int fatherAge;

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

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        this.sonAge = sonAge;

        if (sonAge <= 0) {
            throw new WrongAge(message:"Sons's age cannot be zero or negative");
        }
    }
}
```





```

    }

    if (sonAge >= fatherAge) {
        throw new WrongAge(message:"Son's age cannot be greater than or equal to father's age");
    }
}
}

```

```

public class Main25 {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print(s:"Enter father's age: ");
            int fatherAge = scanner.nextInt();

            System.out.print(s:"Enter son's age: ");
            int sonAge = scanner.nextInt();

            Son son = new Son(fatherAge, sonAge);

            System.out.println("Father's age: " + fatherAge);
            System.out.println("Son's age: " + sonAge);
        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e);
            System.out.println("Exception caught: " + e.getMessage());
        } catch (Exception e) {

```



```
Son son = new Son(fatherAge, sonAge);

System.out.println("Father's age: " + fatherAge);
System.out.println("Son's age: " + sonAge);
} catch (WrongAge e) {
    System.out.println("Exception caught: " + e);
    System.out.println("Exception caught: " + e.getMessage());
} catch (Exception e) {
    System.out.println("Error: " + e);
    System.out.println("Error: " + e.getMessage());
} finally {
    scanner.close();
}
}
```

```
cd "/Users/mrunalini/Desktop/java programs/" && javac Main25.java && java Main25
```

```
● mrunalini@mrunalinis-MacBook-Air java programs % cd "/Users/mrunalini/Desktop/java programs/" && javac Main25.java && java Main25
```

```
Enter father's age: 50
```

```
Enter son's age: -10
```

```
Exception caught: WrongAge: Son's age cannot be negative or zero
```

```
Exception caught: Son's age cannot be negative or zero
```

```
● mrunalini@mrunalinis-MacBook-Air java programs % cd "/Users/mrunalini/Desktop/java programs/" && javac Main25.java && java Main25
```

```
Enter father's age: 30
```

```
Enter son's age: 40
```

```
Exception caught: WrongAge: Son's age cannot be greater than or equal to father's age
```

```
Exception caught: Son's age cannot be greater than or equal to father's age
```

```
○ mrunalini@mrunalinis-MacBook-Air java programs % █
```

