

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

```
import java.util.Scanner;

// Custom exception class WrongAge
class WrongAge extends Exception {
    WrongAge() {
        super("Invalid age. Age cannot be less than 0.");
    }

    WrongAge(String message) {
        super(message);
    }
}

// Base class Father
class Father {
    private int age;

    // Constructor in Father class with age validation
    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge();
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

// Derived class Son extending Father
class Son extends Father {
    private int sonAge;

    // Constructor in Son class with additional age validation
    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge); // Call the constructor of the base class

        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be less than 0.");
        } else if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age should be less than Father's age.");
        }

        this.sonAge = sonAge;
    }
}
```

```

    }

    public int getSonAge() {
        return sonAge;
    }
}

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

        try {
            // Taking user input for Father's age
            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();
            Father father = new Father(fatherAge);
            System.out.println("Father's age: " + father.getAge());

            // Taking user input for Son's age
            System.out.print("Enter Son's age: ");
            int sonAge = scanner.nextInt();
            Son son = new Son(father.getAge(), sonAge);
            System.out.println("Son's age: " + son.getSonAge());

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

    }
}

```

Enter Father's age: 32

Father's age: 32

Enter Son's age: 5

Son's age: 5

Process finished with exit code 0