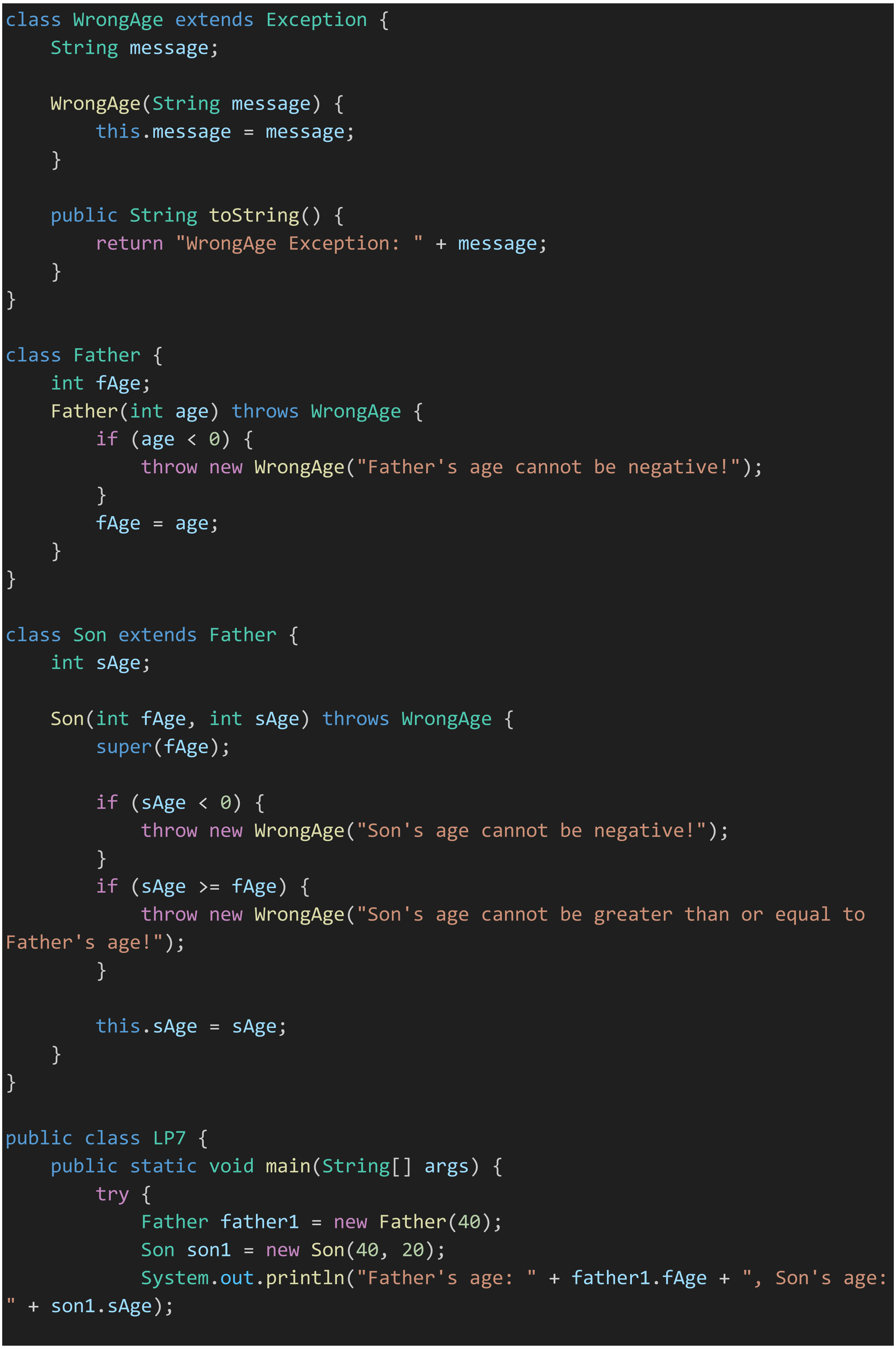
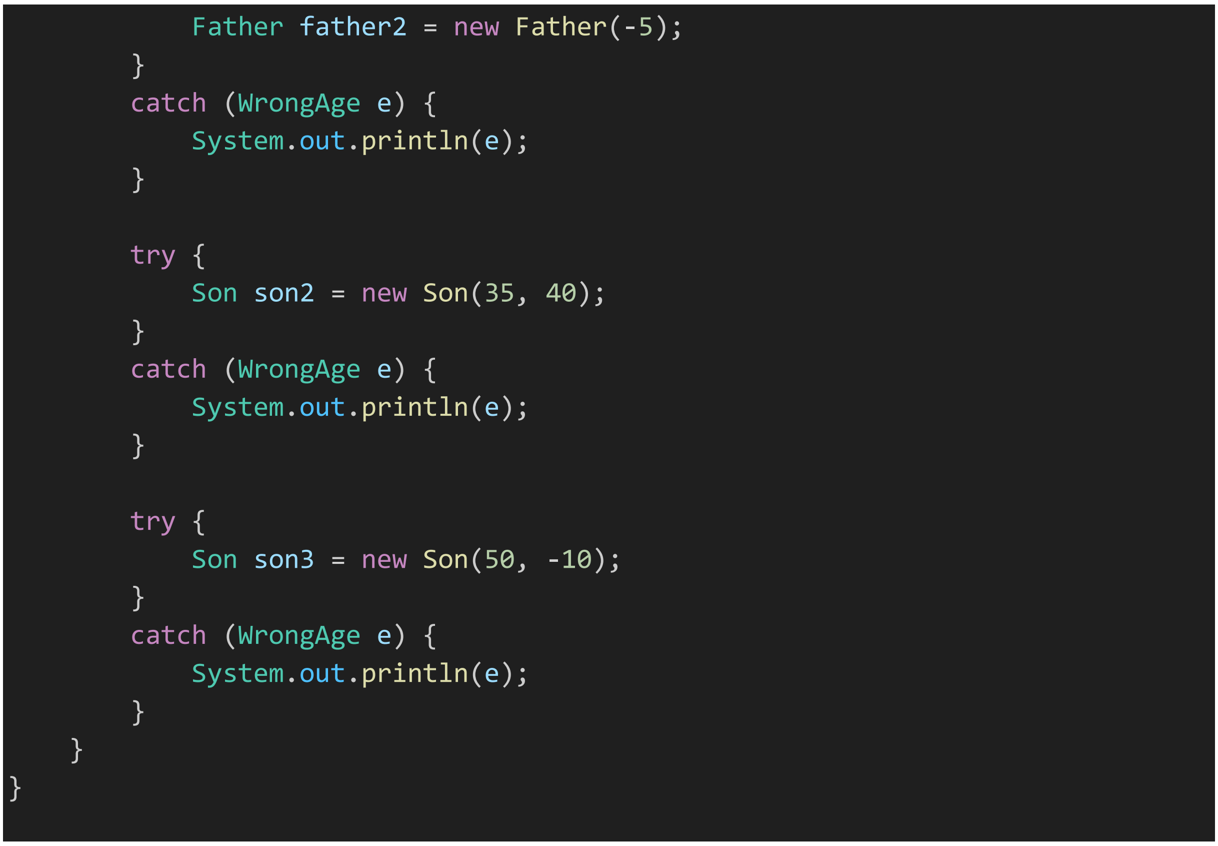
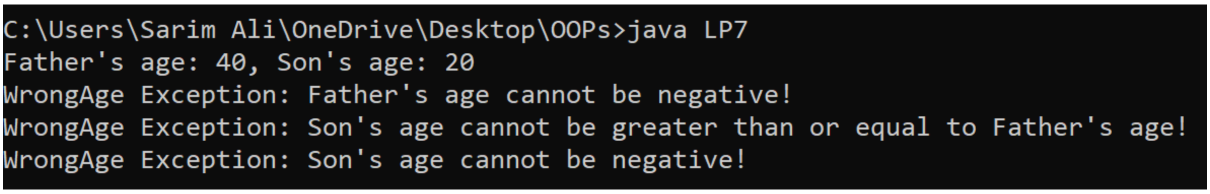


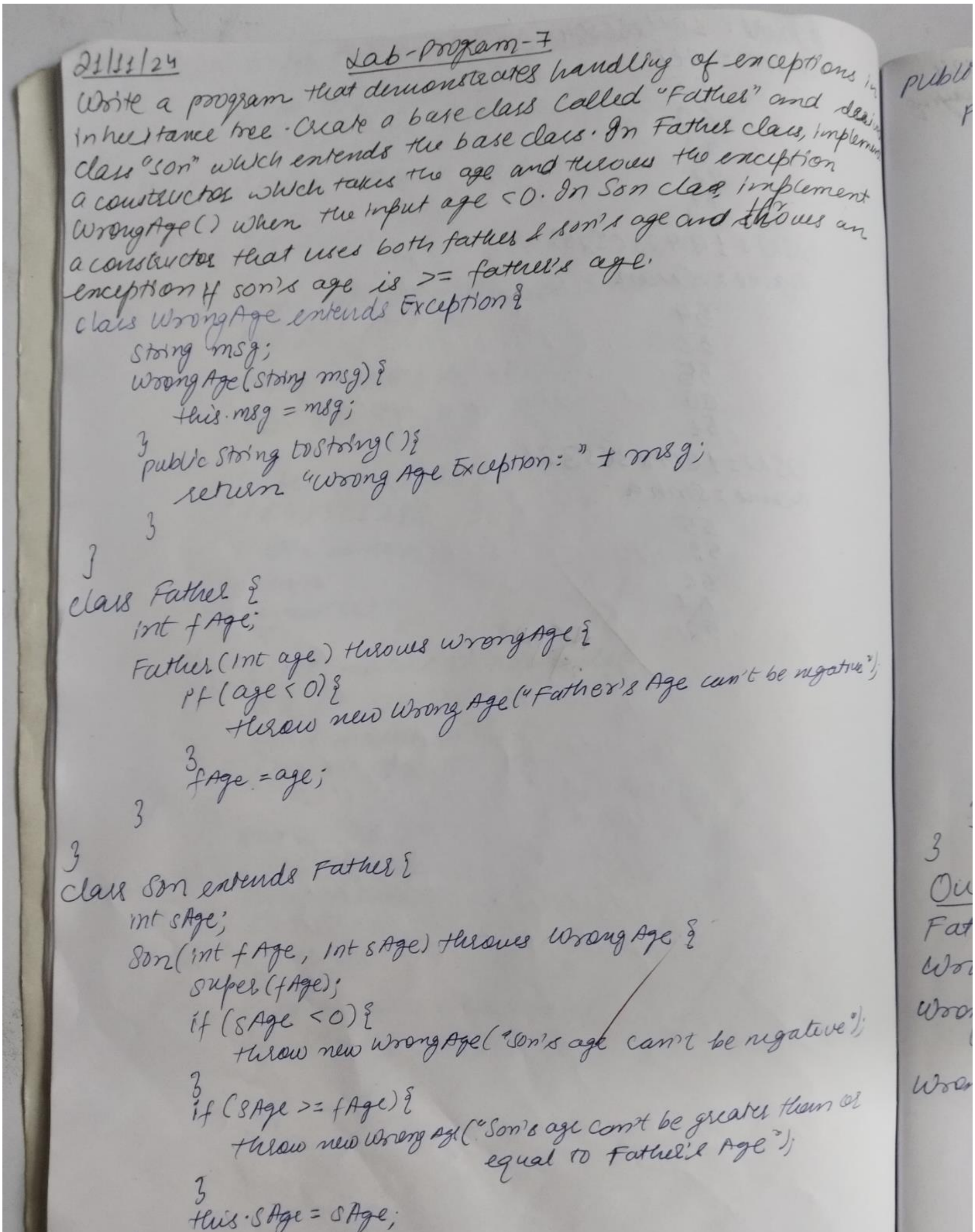
```java
class WrongAge extends Exception {
    String message;

    WrongAge(String message) {
        this.message = message;
    }

    public String toString() {
        return "WrongAge Exception: " + message;
    }
}

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

class Son extends Father {
    int sAge;

    Son(int fAge, int sAge) throws WrongAge {
        super(fAge);

        if (sAge < 0) {
            throw new WrongAge("Son's age cannot be negative!");
        }
        if (sAge >= fAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to
Father's age!");
        }

        this.sAge = sAge;
    }
}

public class LP7 {
    public static void main(String[] args) {
        try {
            Father father1 = new Father(40);
            Son son1 = new Son(40, 20);
            System.out.println("Father's age: " + father1.fAge + ", Son's age:
" + son1.sAge);
```



```java
            Father father2 = new Father(-5);
        }
        catch (WrongAge e) {
            System.out.println(e);
        }

        try {
            Son son2 = new Son(35, 40);
        }
        catch (WrongAge e) {
            System.out.println(e);
        }

        try {
            Son son3 = new Son(50, -10);
        }
        catch (WrongAge e) {
            System.out.println(e);
        }
    }
}
```



```
C:\Users\Sarim Ali\OneDrive\Desktop\OOPs>java LP7
Father's age: 40, Son's age: 20
WrongAge Exception: Father's age cannot be negative!
WrongAge Exception: Son's age cannot be greater than or equal to Father's age!
WrongAge Exception: Son's age cannot be negative!
```



21/11/24 Lab-Program-7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class "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 & son's age and throws an exception if son's age is >= father's age.

```
class WrongAge extends Exception{
    String msg;
    WrongAge(String msg){
        this.msg = msg;
    }
    public String toString(){
        return "Wrong Age Exception: " + msg;
    }
}
class Father {
    int fAge;
    Father(int age) throws WrongAge {
        if(age < 0){
            throw new WrongAge("Father's Age can't be negative");
        }
        fAge =age;
    }
}
class Son extends Father{
    int sAge;
    Son(int fAge, int sAge) throws WrongAge {
        super(fAge);
        if (sAge <0){
            throw new WrongAge("Son's age can't be negative");
        }
        if (sAge >= fAge){
            throw new WrongAge("Son's age can't be greater than or
                                  equal to Father's Age");
        }
        this.sAge = sAge;
    }
}
```

```
public class TryCatch {
    public static void main(String[] args) {
        try {
            Father f1 = new Father(40);
            Son s1 = new Son(40, 20);
            System.out.println("Father's Age:" + f1.fAge + ", Son's
                        Age:" + s1.sAge);
            Father f2 = new Father(-5);
        }
        catch(WrongAge e){
            System.out.println(e);
        }

        try {
            Son s2 = new Son( 35, 40);
        }
        catch{WrongAge e){
            System.out.println(e);
        }
        try {
            Son s3 = new Son(50, -10);
        }
        catch{WrongAge e){
            System.out.println(e);
        }
    }
}
```

Output:-

```
Father's Age: 40, Son's Age: 20
WrongAge Exception: Father's Age can't be negattie
WrongAge Exception:- Son's age can't be greater than or
                  equal to Father's Age.
WrongAge Exception: Son's age cannot be negative
```