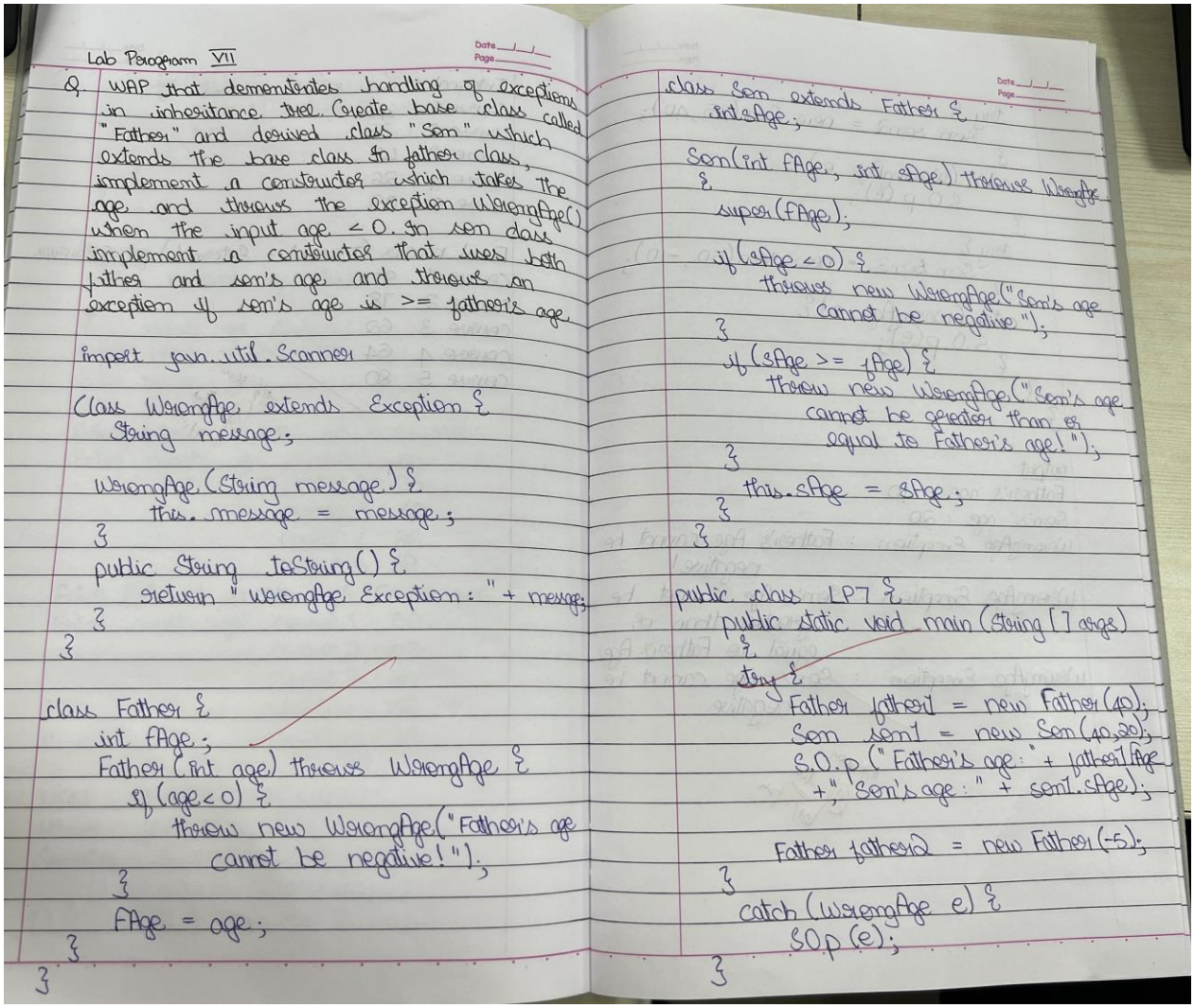
```
class WrongAge extends Exception {
    String message;

    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 FSAGE {
    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);

            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\Admin\Documents\23cs310>javac FSAGE.java

C:\Users\Admin\Documents\23cs310>java FSAGE.java
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!
```



Lab Program VII

Q. WAP that demonstrates handling of exceptions in inheritance tree. Create 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 and son's age and throws an exception if son's age is >= father's age.

```
import java.util.Scanner;

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) {
            throws 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 father = new Father(40);
            Son son = new Son(40,20);
            S.O.p("Father's age: " + father.fAge
                +", Son's age: " + son.sAge);

            Father father2 = new Father(-5);
        }
        catch(WrongAge e) {
            SOp(e);
        }
}
```

```
try {
    Son son2 = new Son(35, 40);
}
catch (WrongAge e) {
    S.O.p(e);
}
try {
    Son son3 = new son(50, -10);
}
catch (WrongAge e){
    S.O.p(e);
}
}
}
```

output

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

o/p seen
21/4/24