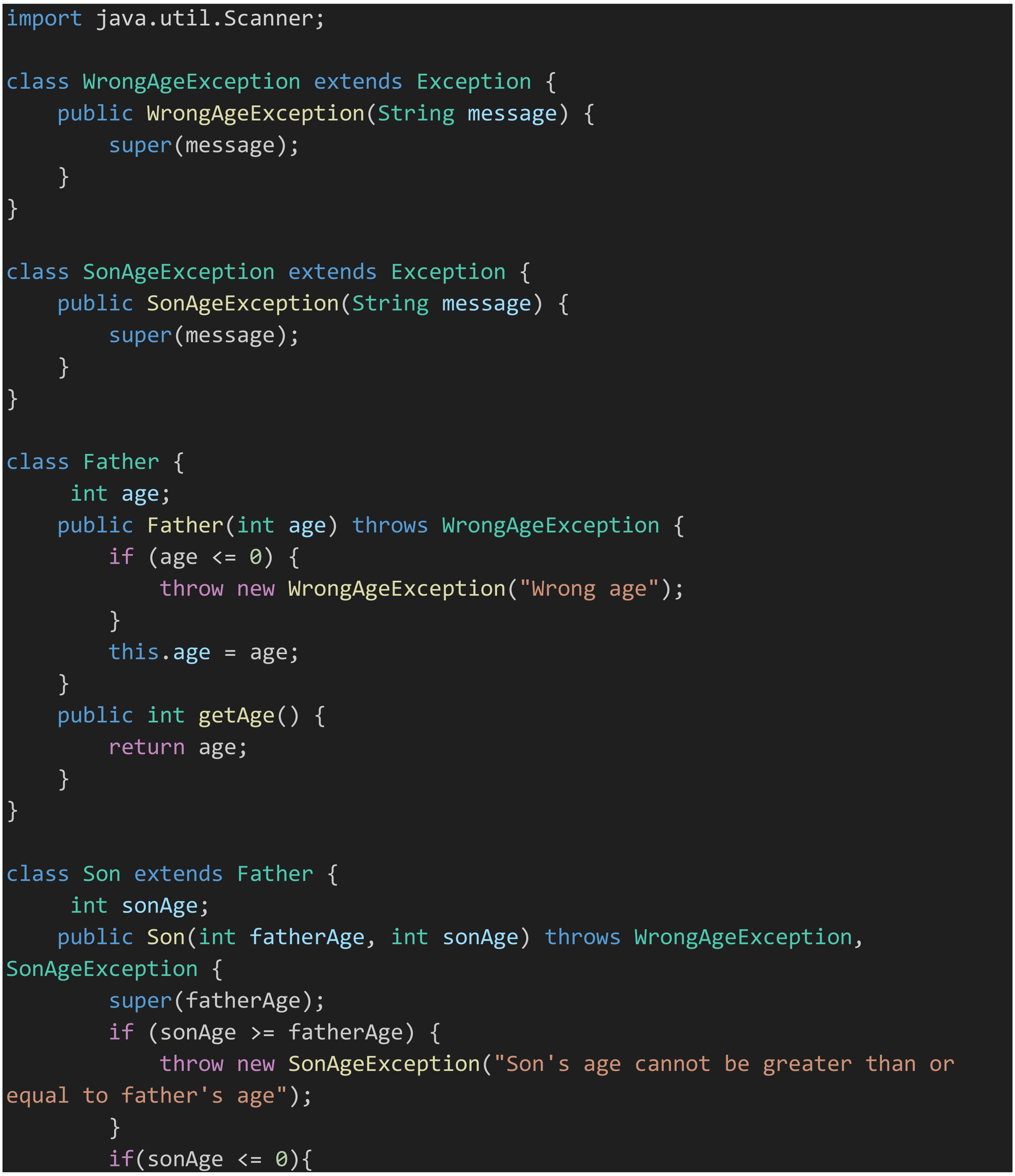
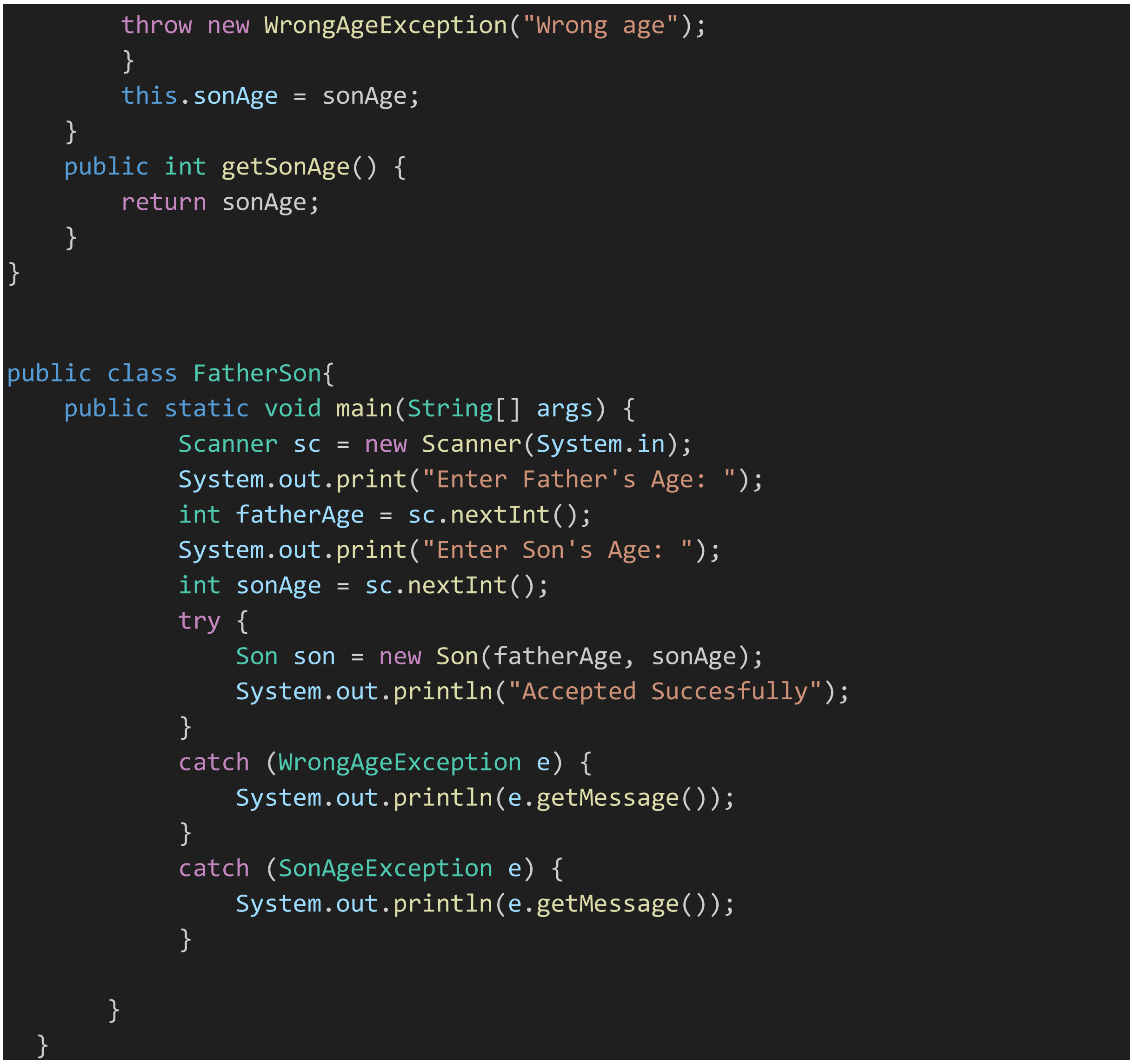
WEEK 7 :

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

Source Code :

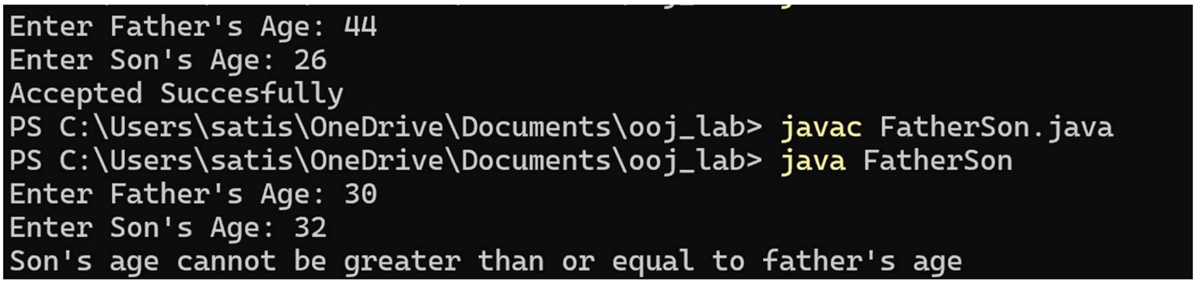


```
import java.util.Scanner;

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

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

class Father {
     int age;
    public Father(int age) throws WrongAgeException {
        if (age <= 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
     int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException,
SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be greater than or
equal to father's age");
        }
        if(sonAge <= 0){
```

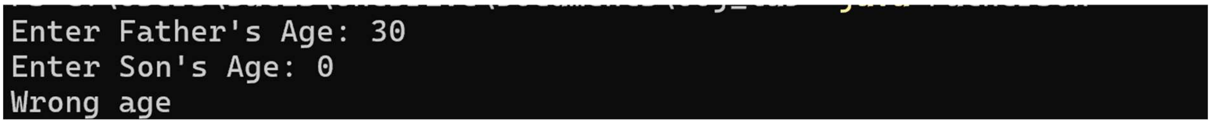


```
        throw new WrongAgeException("Wrong age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}


public class FatherSon{
    public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Father's Age: ");
            int fatherAge = sc.nextInt();
            System.out.print("Enter Son's Age: ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Succesfully");
            }
            catch (WrongAgeException e) {
                System.out.println(e.getMessage());
            }
            catch (SonAgeException e) {
                System.out.println(e.getMessage());
            }

       }
  }
```

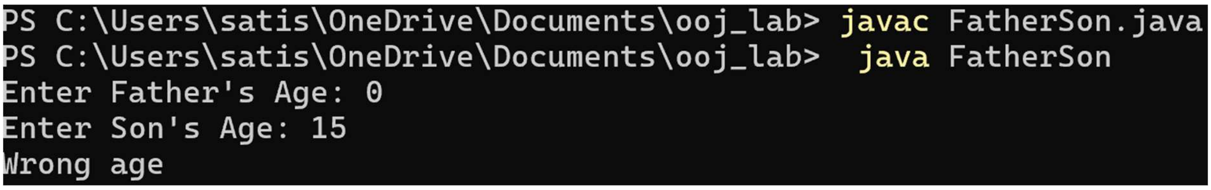
Output :



```
Enter Father's Age: 44
Enter Son's Age: 26
Accepted Succesfully
PS C:\Users\satis\OneDrive\Documents\ooj_lab> javac FatherSon.java
PS C:\Users\satis\OneDrive\Documents\ooj_lab> java FatherSon
Enter Father's Age: 30
Enter Son's Age: 32
Son's age cannot be greater than or equal to father's age
```

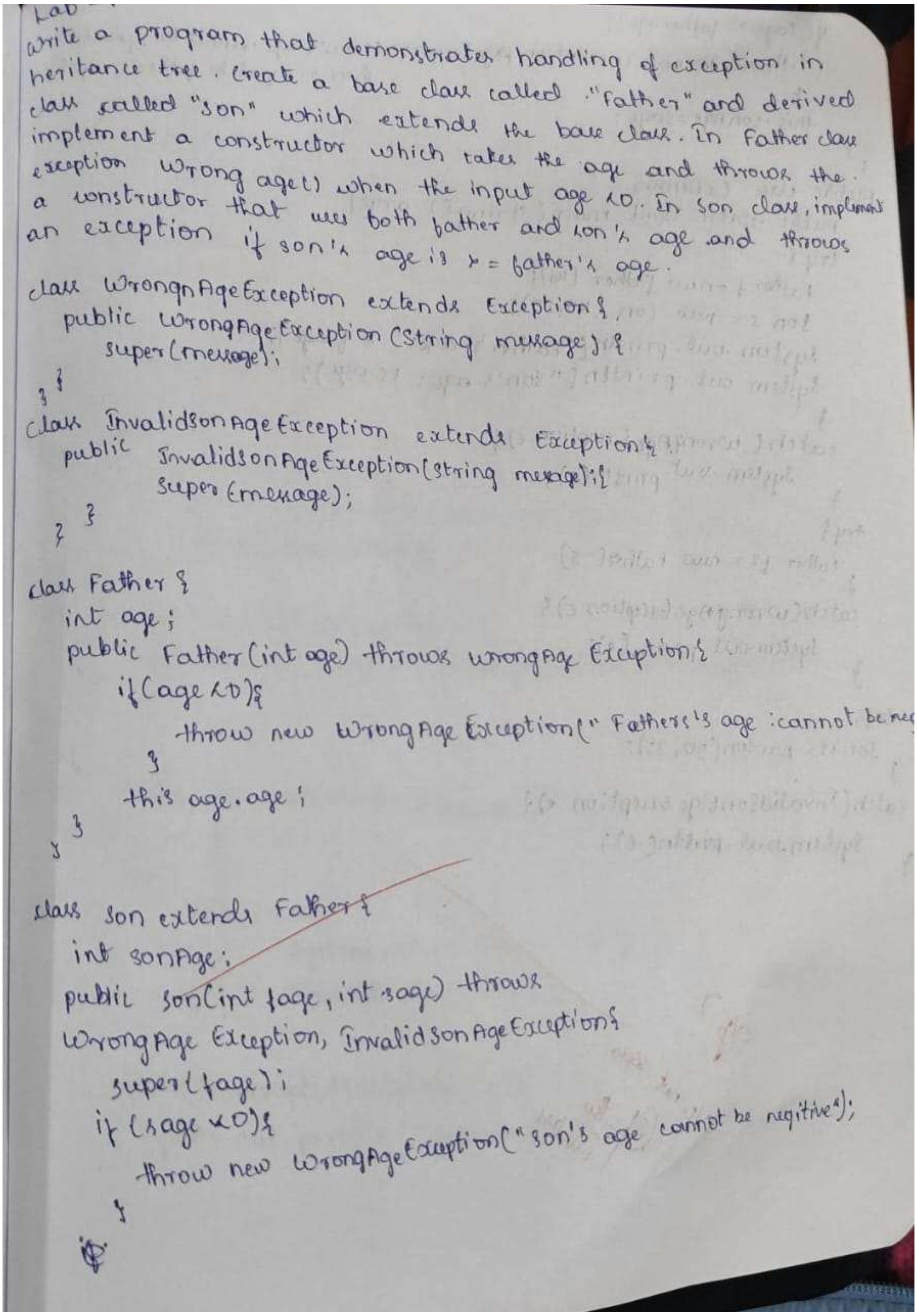


```
Enter Father's Age: 30
Enter Son's Age: 0
Wrong age
```



```
PS C:\Users\satis\OneDrive\Documents\ooj_lab> javac FatherSon.java
PS C:\Users\satis\OneDrive\Documents\ooj_lab>  java FatherSon
Enter Father's Age: 0
Enter Son's Age: 15
Wrong age
```

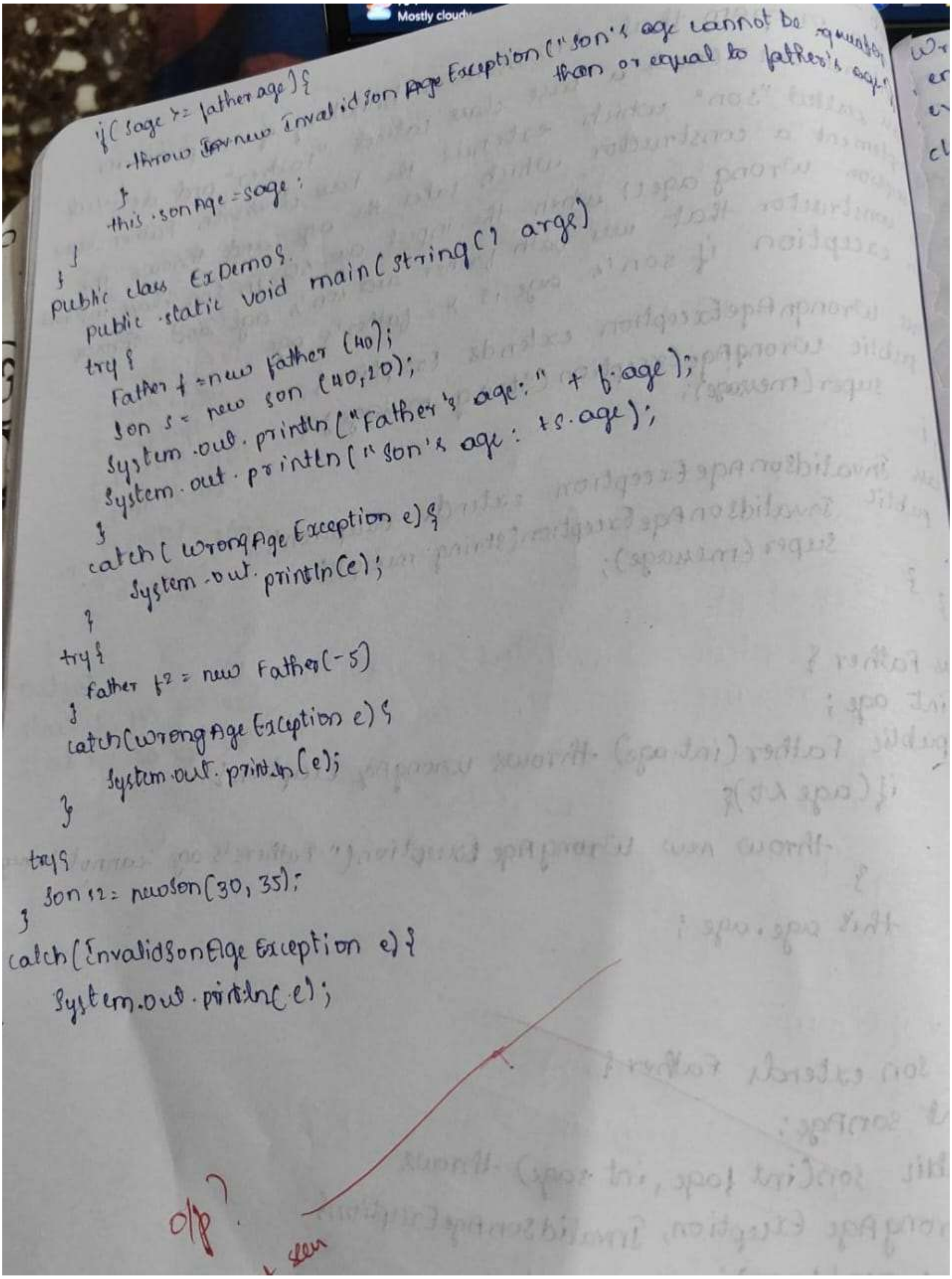
Written Code & Output :



Lab

write a program that demonstrates handling of exception in
heritance 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 Wrong age() when the input age <0. In son class, implements
a constructor that uses both father and son's age and throws
an exception if son's age is >= father's age.

```
class WrongnAgeException extends Exception{
   public WrongAgeException (String message ){
      super(message);
  }
}
class InvalidSonAgeException extends Exception{
   public    InvalidSonAgeException(string message);{
             Super (message);
    }
 }

class Father {
   int age;
   public Father(int age) throws wrongAge Exception{
       if(age <0){
           throw new WrongAge Exception(" Fathers's age :cannot be ne
        }
       this age.age ;
   }
 }

class son extends Father{
   int sonAge;
   public son(int fage, int sage) throws
   WrongAge Exception, InvalidSonAgeException{
      super(fage);
     if (sage <0){
        throw new WrongAgeException("son's age cannot be negitive");
     }
```



```
if( sage >= father age ){
   throw new InvalidSonAgeException("Son's age cannot be greater
                                     than or equal to father's age")
   }
   this.sonAge = sage;
 }
}
public class ExDemo{
  public static void main(String[] arge)
  try{
    Father f = new Father (40);
    Son s = new Son (40,20);
    System.out.println("Father's age:" + f.age);
    System.out.println("Son's age: +s.age);
  }
  catch( WrongAge Exception e){
      System.out.println(e);
  }
 try{
   Father f2 = new Father(-5)
 }
 catch(WrongAgeException e){
     System.out.println(e);
 }
try{
 Son s2= new Son(30, 35);
}
catch(InvalidSonAge Exception e){
  System.out.println(e);
```

o/p ?

seen