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

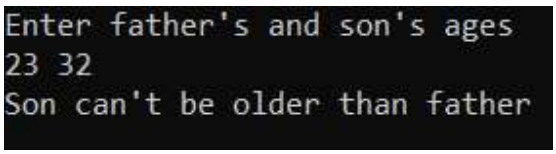
```
import java.util.Scanner;

class WrongAgeException extends Exception {
    public String toString() {
        return ("Negative age can't be accepted");
    }
}

class AgeException extends Exception {
    public String toString() {
        return ("Son can't be older than father");
    }
}

class Father {
    int father_age;

    Father(int x) throws WrongAgeException {
        father_age = x;
        if (father_age < 0) {
            throw new WrongAgeException();
        }

    }
}

class Son extends Father {
    int son_age;

    Son(int x, int y) throws AgeException, WrongAgeException {
        super(x);
        son_age = y;
        if (son_age < 0) {
```

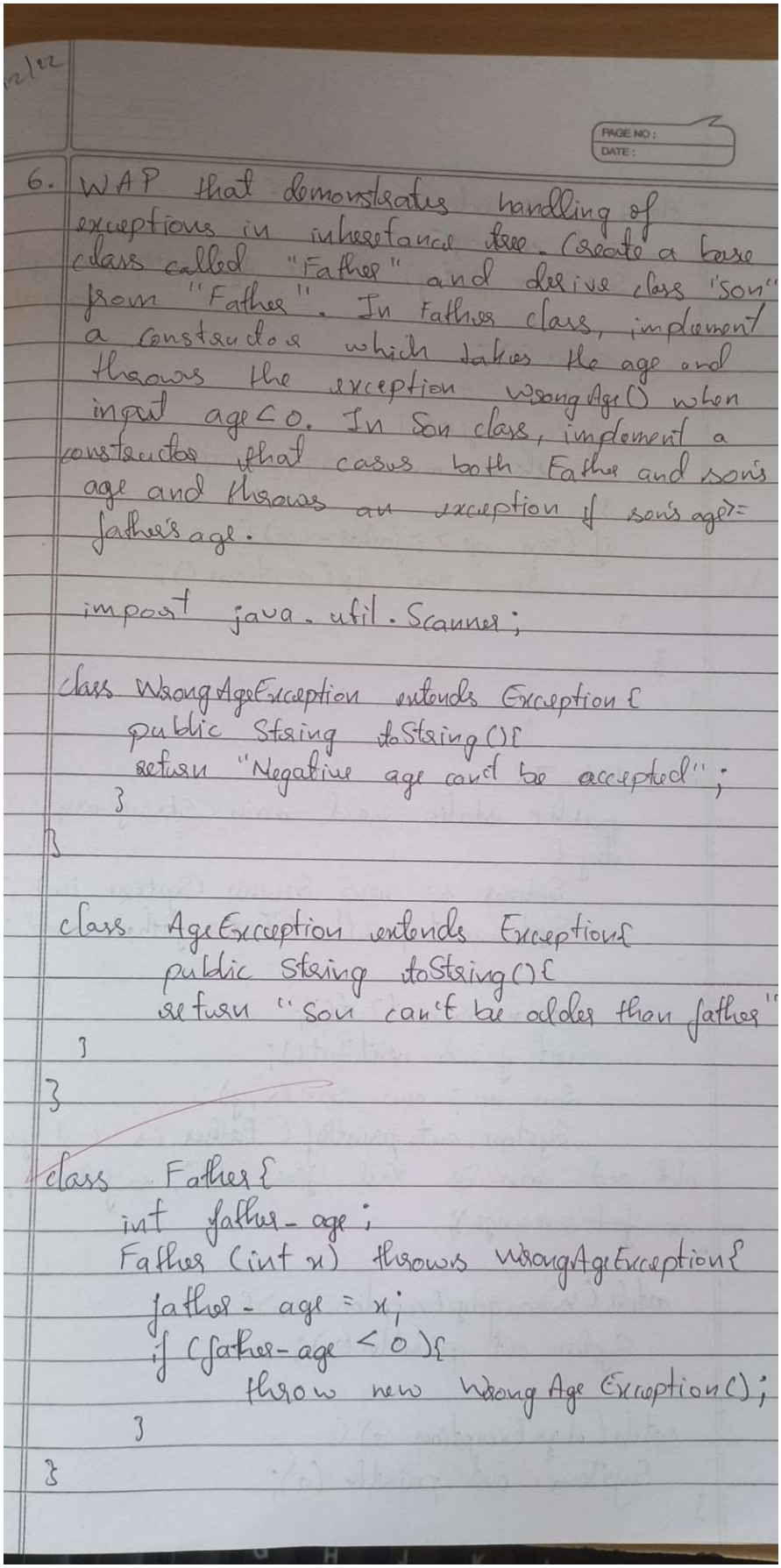
```
            throw new WrongAgeException();
        }
        if (son_age >= father_age) {
            throw new AgeException();
        }
    }
}

class Lab_7 {
    public static void main(String[] args) {
        try {
            Scanner s = new Scanner(System.in);
            System.out.println("Enter father's and son's ages");
            int x = s.nextInt();
            int y = s.nextInt();
            Son so = new Son(x, y);
            System.out.printf("Father is %d years old and son is %d years old", so.father_age,
so.son_age);
        } catch (WrongAgeException wa) {
            System.out.println(wa);
        } catch (AgeException a) {
            System.out.println(a);
        } catch (Exception e) {
            System.out.println("Enter valid values");
        }
    }
}
```

```
Enter father's and son's ages
35 15
Father is 35 years old and son is 15 years old
```

```
Enter father's and son's ages
-98 90
Negative age can't be accepted
```

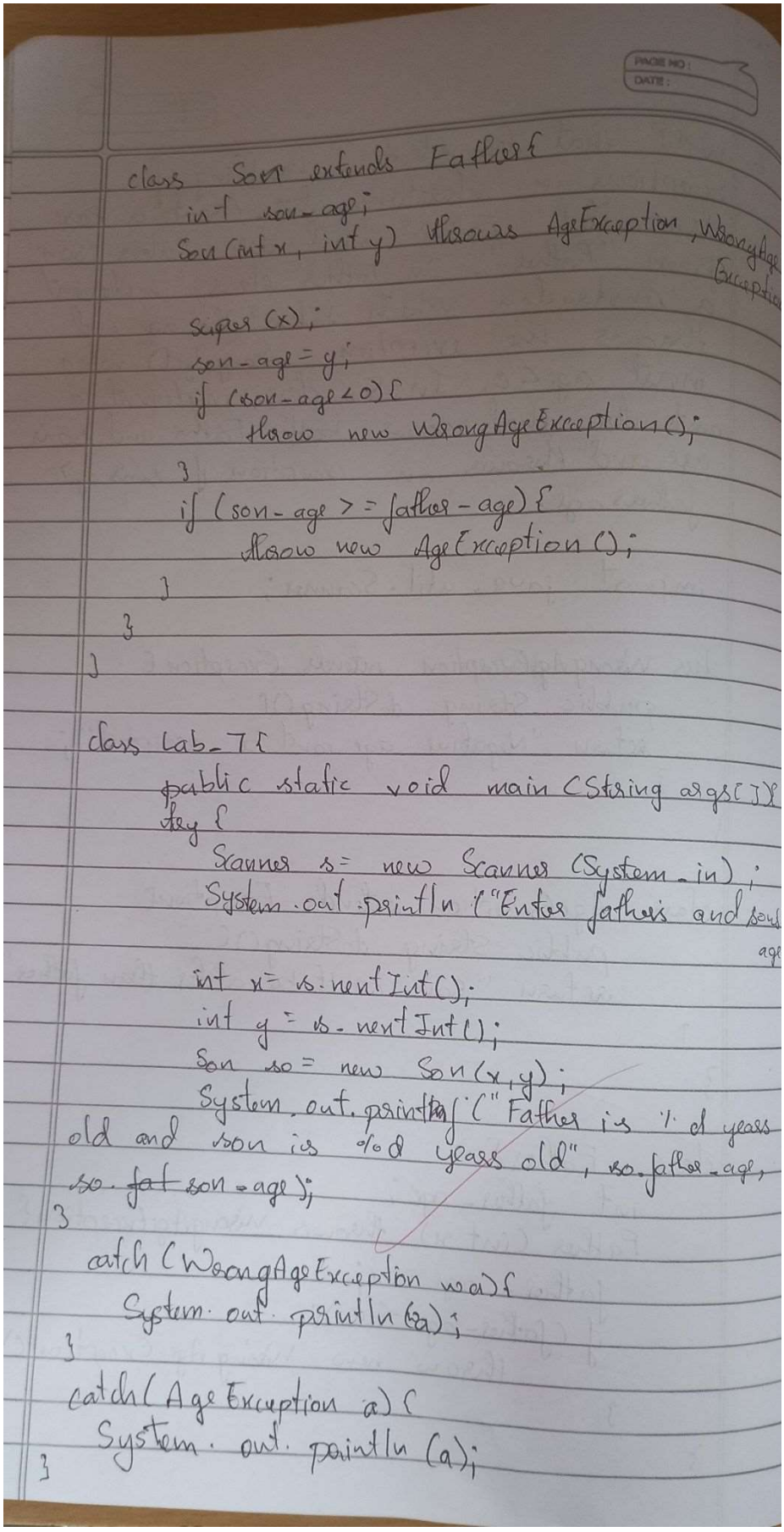
```
Enter father's and son's ages
34 -9
Negative age can't be accepted
```


Enter father's and son's ages
23 32
Son can't be older than father

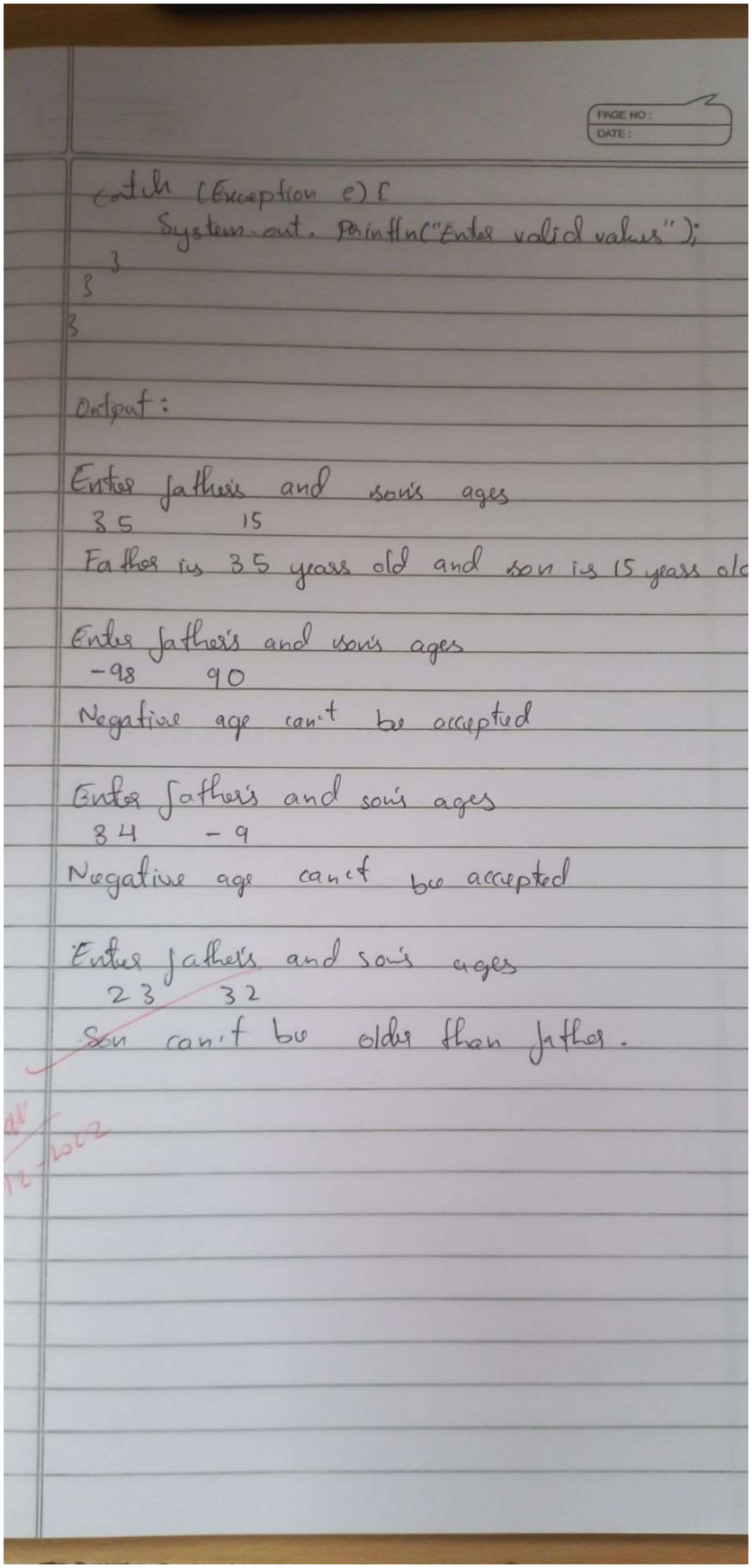


6. WAP that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derive class "Son" from "Father". In Father class, implement a constructor which takes the age and throws the exception WrongAge() when input age < 0. In Son class, implement a constructor that cases both Father and Son's age and throws an exception if son's age >= father's age.

```
import java.util.Scanner;

class WrongAgeException extends Exception {
    public String toString(){
    return "Negative age can't be accepted";
    }
}

class AgeException extends Exception{
    public String toString(){
    return "Son can't be older than father";
    }
}

class Father{
    int father_age;
    Father(int x) throws WrongAgeException{
        father_age = x;
        if(father_age < 0){
            throw new WrongAgeException();
        }
}
```



```
class Son extends Father{
    int son_age;
    Son(int x, int y) throws AgeException, WrongAge
                                          Exception
    {
        super(x);
        son_age = y;
        if (son_age < 0){
            throw new WrongAgeException();
        }
        if (son_age >= father_age){
            throw new AgeException();
        }
    }
}

class Lab_7{
    public static void main(String args[])
    try{
        Scanner s = new Scanner(System.in);
        System.out.println("Enter father's and son's
                                              age");
        int x = s.nextInt();
        int y = s.nextInt();
        Son so = new Son(x,y);
        System.out.printf("Father is %d of years
old and son is %d years old", so.father_age,
so.~~fat~~son_age);
    }
    catch(WrongAgeException wa){
        System.out.println(wa);
    }
    catch(AgeException a){
        System.out.println(a);
    }
```



```
catch (Exception e){
    System.out.println("Enter valid values");
}
}
```

Output:

```
Enter fathers and sons ages
35 15
Father is 35 years old and son is 15 years old

Enter father's and son's ages
-98 90
Negative age can't be accepted

Enter father's and son's ages
34 -9
Negative age can't be accepted

Enter father's and son's ages
23 32
Son can't be older than father.
```