(

LAB PROGRAM VV
The state of the state of
WAITE a proposition of the same
"Father" and derived class of
"Father" and derived class called
"Father" and derived class called "Son" which extend the
which takes the age and the
which takes the age and throws reverption wrong Age() when the input age to In son class implement a constructor that uses father and son's age and throws an econstructor
that uses father and in a
1 son's age >= father's age
relass wrong age extends Exception
& Super (Messer)
2 subox (wessade);
(0)
3
class Father of
father Age
public Father (int age) throws wrong Age of
of (age < o) d
O / a
throw new Wrong Age ("Father's age, can't be
throw new Wrong Age ("Father's age, can't be
negative);
negative);
throw new Wrong Age ("Father's age, can't be
negative);
throw new Wrong Age ("Father's age can't be negative"); this. father Age = age;
throw new Wrong Age ("Father's age can't be negative"); this. father Age = age;

DATE	1	1,	- N
PAGE		ı	\leq

Output: Sold promote the promote of		DATE / / PAGE
Son Age can't be greaten than father's Age Father's Age 50 - Son's Age mount in the stand Committee the greaten than father's Age Father's Age 50 - Son's Age mount in the stand Committee the greaten than father's Age (Committee the greaten the greaten than father's Age (Committee the greaten the greaten than father's Age (Committee the greaten t	a see a	CAB PRINCES & STALL
Son Age can't bell negative was a son Age Fatherin Age son's Age	D-1 174	ett put
Son Age can't bell negative was a too a get too a greater than father's Age Father's Age son's		Enter internal morte many
Son age can't be greater than father's age Father's Age is a greater than father's age Fother's Age is a greater than father's age The constant and the greater than father age The constant and the greater than a material age The constant and the greater than a greater th		activity of the state of the st
Father's Age some Son's Age substitute as a line Formal aide The control of the south The con		Son Age can't best negative
Taken Age = 50 pt Son's Age processor and in the son in		or have can't be greater than fother's Age
Const from the freeze of the format of the f		Father's Age 5 Ch Son's
Frank Stand with the stand of t	Asia	Age Trascique to section
So,25); The desired state material of the state of the s		The west
So,25); Le designate integrant to date and		Trans of the Dist
So, 25); So binned standard to to make the second of the		September of the motor of the september
So,25); b hand charter 320 intend 22d intend 22d intend 22d intend 22d intend 22d intend 22d intended 22d in		Janes Lawrell of
So,25); b hand charter 320 intend 22d intend 22d intend 22d intend 22d intend 22d intend 22d intended 22d in		
So,25); b hand shooter 720 interes and red red red red red red red red red re		catch of Totam. place Europeion a. 1
Public and runts Throat of public and runts & public and runts & public (mar) of the (mar) of		morganistas transfer 11) tim?
hort. hort. hort. hort. hort. hort.	50,25);	E T
To (mer) of drag		
hort. hort. hort. hort. hort. hort.		
To (mer) of drag		the ancolaitet contents thronk of
((and all)		De Jane King, allowy
(("aya") +un2		
e (consinged a house		
		e / coot costs known

lab 7

```
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}
class Father {
    int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Father's age cannot be negative.");
        this.age = age;
   }
}
class Son extends Father {
    int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equ
        this.sonAge = sonAge;
    }
}
public class InheritanceExceptionDemo {
    public static void main(String[] args) {
        try {
            Father father = new Father(40);
            Son son = new Son(40, 20);
            System.out.println("Father's age: " + father.age);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
        try {
```

```
Father father = new Father(-5);
} catch (WrongAgeException e) {
    System.out.println("Exception: " + e.getMessage());
}

try {
    Son son = new Son(30, 30);
} catch (WrongAgeException e) {
    System.out.println("Exception: " + e.getMessage());
}
}
```

▼ Output

```
Father's age: 40
Son's age: 20
Exception: Father's age cannot be negative.
Exception: Son's age cannot be greater than or equal to Father's age.
```

lab 7