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.

```java
import java.util.Scanner;
class WrongAgeException extends Exception
{
    WrongAgeException()
    {
        System.out.println("Age cannot be negative");
    }
}
class WrongSonAgeException extends Exception
{
    WrongSonAgeException()
    {
        System.out.println("son's age cannot be great than father");
    }
}
class father
{
    int fage;
    Scanner ss = new Scanner(System.in);
    Father() throws WrongAgeException
    {
        System.out.println("enter father's age");
        fage = ss.nextInt();
```

```java
        if (fage < 0)
        {
            throw new WrongAgeException();
        }
    }
    System.out.println("Father's age is" + fage);
    }
}

class Son extends Father;
{
    int sage;
    Scanner ss = new Scanner(System.in);
    Son() throws WrongSonAgeException, WrongAgeException
    {
        super();
        System.out.println("Enter son's age:");
        sage = ss.nextInt();
        if (sage < 0)
        {
            throw new WrongAgeException();
        }
        if (sage >= fage)
        {
            throw new WrongSonAgeException();
        }
        System.out.println("son's age is " + sage);
    }
}

class Pgmdemo
{
    public static void main(String args[])
    {
        try
        {
            Son s = new Son();
        }
```

```
    catch (Exception e)
    {
        System.out.println(" ages are not given correct");
    }
    }
}
}
```

Output 1: Enter father's age:
        45
        Father's age is 45.
        Enter son's age:
        23
        Son's age is 23

Output 2: Enter father's age:
        - 39
        Age cannot be negative.
        Ages eare not given correct.

Output: 3: Enter father's age:
        34
        Enter father's age is 34
        Enter son's age:
        42
        Son's age cannot be greater than father
        Ages are not given correct.

```java
import java.util.Scanner;
class WrongIntException extends Exception
{
WrongIntException()
{
System.out.println("Internal marks exceeded");
}
}
class WrongExtException extends Exception
{
WrongExtException()
{
System.out.println("External marks exceeded");
}
}
class Internals
{
int im;
Scanner ss=new Scanner(System.in);
Internals() throws WrongIntException
{
System.out.println("enter internal marks:");
im=ss.nextInt();
if (im>30)
{
throw new WrongIntException();
}
System.out.println("Internal marks are "+im);
}
}
class Externals
{
int em;
Scanner sa=new Scanner(System.in);
Externals() throws WrongExtException
{
System.out.println("enter external marks:");
em=sa.nextInt();
if (em>70)
{
throw new WrongExtException();
}
System.out.println("External marks are "+em);
}
}
class Marks
{
public static void main(String args[])
```

```java
{
try
{
Internals i=new Internals();
Externals e=new Externals();
}
catch(Exception e)
{
System.out.println("marks entered are wrong");
}
}
}
```

```
C:\1BM23CS303>javac Pgmdemo.java

C:\1BM23CS303>java Pgmdemo
enter father age:
45
Father's age is 45
enter son's age:
23
Son's age is 23

C:\1BM23CS303>javac Pgmdemo.java

C:\1BM23CS303>java Pgmdemo
enter father age:
-39
Age cannot be negative
ages are not given correct

C:\1BM23CS303>javac Pgmdemo.java

C:\1BM23CS303>java Pgmdemo
enter father age:
34
Father's age is 34
enter son's age:
42
Son's age cannot be greater than father
ages are not given correct
```

```
ages are not given correct

C:\1BM23CS303>javac Pgmdemo.java

C:\1BM23CS303>java Pgmdemo
enter father age:
48
Father's age is 48
enter son's age:
-34
Age cannot be negative
ages are not given correct

C:\1BM23CS303>
```