LAB PROGRAM-6

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.*;


class fatherAgeException extends Exception
{
  public String toString(){
      return("Father's age is less that 0");
  }
}

class sonAgeException extends Exception{
    int a;
    sonAgeException(int age){
        a=age;
    }
    public String toString(){
        if(a<0)
          return("Son's age is less than 0");
        else
          return("Son's age is more than father's age");
    }
}


class Father{
    int age;
    Scanner in=new Scanner(System.in);
    Father(){
        System.out.println("Enter the father's age: ");
        age=in.nextInt();
    }
    void ex1() throws fatherAgeException
    {
        if(age<0)
         throw new fatherAgeException();
    }
```

```
}
class Son extends Father{
    int age;
  Son(){
    System.out.println("Enter the age of son: ");
    age=in.nextInt();
  }
  void ex2() throws sonAgeException{
      if(age<0||age>super.age){
         throw new sonAgeException(age);
      }
  }
}

public class except {
    public static void main(String[] args){
        Son s=new Son();
        try{
            s.ex1();
        }
        catch(fatherAgeException e){
            System.out.println(e);
        }
        try{
            s.ex2();
        }
        catch(sonAgeException e){
            System.out.println(e);
        }
    }
}
```

Program-6

Exceptions

```
import java.util.*;
Class fatherageException extends Exception
{
   public String toString(){
     return ("Fathers age is less than 0");
     }
}
Class sonAgeException extends Exception{
   int a;
   SonAgeException(int age) {
     a=age;
   }
   Public String toString(){
     if (a<0)
     return("Son's age is less than 0");
     else
     return ("Sons age is more than fathes age");
   }
}
Class Father{
     int age
   Scanner in = new Scanner(System.in);
   Father() {
     System.out.println("Enter the Father's age:");
```

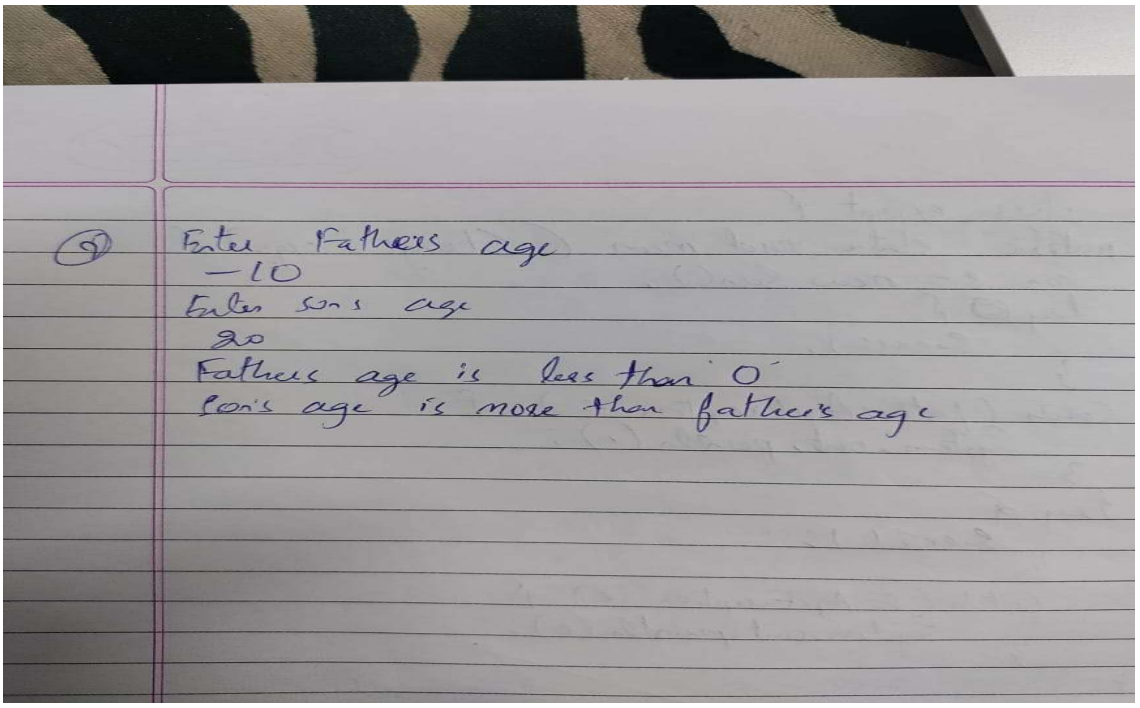
```
        age = in.nextInt();
    }
    void ch1() throws FatherAgeException
    {
        if (age<0)
        throw new FatherAgeException();
    }

}
class Son extends Father {
    int age;
    Son(){
        System.out.println("Enter the age of son:");
        age = in.nextInt();
    }
    void ch2() throws SonAgeException {
        if (age<0 || age>super.age) {
            throw new SonAgeException(age);
        }
    }
}
```

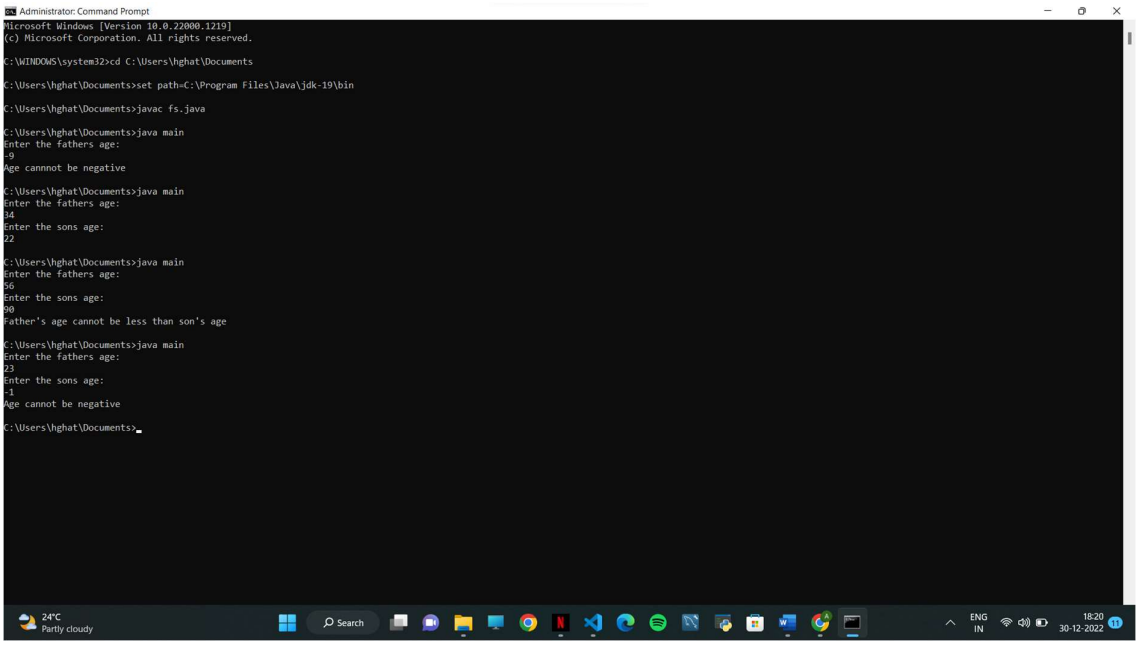
```
Public class except {
    public static void main (String[] args) {
     Son s = new Son();
     try {
         s.ex1();
     }
     Catch (FatherAgeException e) {
         System.out.println (e);
     }
     try {
         s.ex2();
     }
     Catch (SonAgeException e) {
         System.out.println (e);
     }
    }
}
```

30/12/2022

Output

Enter father's age
56
Enter son's age
19


Enter Fathers age
-10
Enter sons age
20
Fathers age is less than 0
Son's age is more than father's age


Administrator: Command Prompt
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>cd C:\Users\hghat\Documents
C:\Users\hghat\Documents>set path=C:\Program Files\Java\jdk-19\bin
C:\Users\hghat\Documents>javac fs.java
C:\Users\hghat\Documents>java main
Enter the fathers age:
-9
Age cannnot be negative
C:\Users\hghat\Documents>java main
Enter the fathers age:
34
Enter the sons age:
22
C:\Users\hghat\Documents>java main
Enter the fathers age:
56
Enter the sons age:
90
Father's age cannot be less than son's age
C:\Users\hghat\Documents>java main
Enter the fathers age:
23
Enter the sons age:
-1
Age cannot be negative
C:\Users\hghat\Documents>
24°C
Partly cloudy
Search
ENG
IN
18:20
30-12-2022