NAME: Ramya Ramesh

USN: 1BM19CH038

**LAB PROGRAM 8**

QUESTION: 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.

CODE:

import java.util.Scanner;

class WrongAge extends Exception {

int fatherAge;

WrongAge(int fAge)

{

this.fatherAge=fAge;

}

public String toString()

{

return("\n ERROR: Father's age can't be negative!");

}

}

class SonException extends Exception

{

int f,s;

SonException(int fAge,int sAge)

{

this.f=fAge;

this.s=sAge;

}

public String toString()

{

if(f==s)

return("\n ERROR: Son's age can't be equal to father's age!");

if(s<0)

return("\n ERROR: Son's age can't be less than zero!");

else

return("\n ERROR: Son's age can't be greater than father's age!");

}

}

class Father

{

int fAge;

Scanner sc=new Scanner(System.in);

Father()

{

System.out.println("Enter the father's age:\n");

fAge=sc.nextInt();

}

void exception1() throws WrongAge

{

if(fAge<0)

throw new WrongAge(fAge);

}

}

class Son extends Father

{

int sAge;

Scanner sc=new Scanner(System.in);

Son()

{

super();

System.out.println("Enter the son's age:\n");

sAge=sc.nextInt();

}

void exception2() throws SonException

{

if(sAge<0 || sAge>=fAge)

throw new SonException(fAge,sAge);

}

}

class except

{

public static void main(String args[])

{

Son s =new Son();

try{

s.exception1();

}

catch(WrongAge e)

{

System.out.println("Exception caught" + e);

}

try{

s.exception2();

}

catch(SonException e)

{

System.out.println("Exception caught" + e);

}

}

}

OUTPUT:

