

Lab - 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 takes both father and son's age and throws an exception if son's age is >= father's age.

```
import java.util.Scanner;  
class WrongAge extends Exception  
{  
    public WrongAge(String message)  
    {  
        super(message);  
    }  
}
```

```
class InputScanner  
{  
    protected Scanner s;  
    public InputScanner()  
    {  
        s = new Scanner(System.in);  
    }  
}
```

class Father extends InputScanner

{
protected int fatherAge;

public Father() throws WrongAge

{
System.out.println("Enter father's
age:");
fatherAge = s.nextInt();

if (fatherAge < 0)

{
throws new WrongAge("Age
cannot be negative");

public void display()

{
System.out.println("Father's
Age: " + fatherAge);

class Son extends Father

{
private int sonAge;

public Son() throws WrongAge

{
super();

System.out.println("Enter son's
age:");
sonAge = s.nextInt();

{ if (sonAge >= fatherAge)

 throw new WrongAge("Son's age
 cannot be greater than father
 age");

}

else if (sonAge < 0)

{

 throw new WrongAge("Son's
 Age: " + sonAge);

}

public class Main

public static void main(String[] args)

{ try

 Son son = new Son();
 son.display();

}

catch (WrongAge e)

 System.out.println("Error: " +
 e.getMessage());

}

3

3

Output

Enter father's age:

45

Enter son's age:

20

Father's Age: 45

Son's Age: 20

Enter father's age:

25

Enter son's age:

50

Error: Son's age cannot be greater than
father's age.

Sakshi, B.R

IBM02CS233