LAB

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, implements a constructor which takes the age and throws 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 >= father's age.

```
import java.util.*;
class WrongAge extends Exception {
  public WrongAge(String message){
    super(message);
  }
}
class Father {
  int fage;
  public Father(int age) throws WrongAge {
  if (age<0)
  throw new WrongAge("Father age cannot be negative");
  this.fage = age;
  System.out.println("Father age : " + age);
}
class Son extends Father {
  int sonAge;
public Son(int fage, int sonAge) throws WrongAge{
  super(fage);
```

```
if (sonAge < 0){
throw new WrongAge("Son age cannot be negative");
}
if (sonAge >= fage){
throw new WrongAge("Son's age cannot be greater
                    or equal to father age");
}
this.sonAge = sonAge;
System.out.println("Son age : " + sonAge);
}
}

public class InheritanceExceptionDemo {
public static void main(String args[]){
try {
Father f = new Father(45);
Son s = new Son(45, 20);
}
catch (WrongAge e) {
System.out.println("Exception caught : " + e.getMessage());
}
try {
Father f = new Father(-5);
}
catch (WrongAge e) {
System.out.println("Exception caught: " + e.getMessage());
}

try {
Son s = new Son(40, 50);
}
catch (WrongAge e) {
```

```
if (sonAge < 0){
throw new WrongAge("Son age cannot be negative");
}
if (sonAge >= fage){
throw new WrongAge("Son's age cannot be greater
                    or equal to father age");
}
  this.sonAge = sonAge;
  System.out.println("Son age : " + sonAge);
}

public class InheritanceExceptionDemo {
public static void main (String args[]){
try {
  Father f = new Father(45);
  Son s = new Son(45, 20);
}
catch (WrongAge e) {
  System.out.println("Exception caught : " + e.getMessage());
}
try {
  Father f = new Father(-5);
}
catch (WrongAge e) {
  System.out.println("Exception caught: " + e.getMessage());
}

try {
  Son s = new Son(40, 50);
}
catch (WrongAge e){
  System.out.println("Exception caught: " + e.getMessage());
```

Output:

Father's age : 45
Father's age : 45
Son's age : 20
Exception caught : Father's age cannot be negative
Father's age : 40
Exception caught : Son's age cannot be greater than or equal to Father's age

Extra Programs

1) Arithematic Exception ⟹ / by zero

2) File : test.txt is missing. Please check exception prog dpt file name

3) Please enter your age - Numeric value
11
you are not authorized
64
you are authorized.

4) java.lang.ArithematicException : / by zero
at GFG1.main(GFG1.java:9)

java.lang.ArithematicException : / by zero

java.lang.ArithematicException : / by zero

Type an integer
23
you typed 23

as
rapping Exception