

## Program - 7

Q. Write a program that demonstrates handling of exception in inheritance tree. Create a base class called "Father" and derived class "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 & son's age and throws an exception if son's age is  $\geq$  father's age.

Ans  
Import java.util.Scanner;  
class WrongAgeException extends Exception {  
 public WrongAgeException (String message) {  
 super(message);  
 }  
}

Class Father {

private int age ;

public Father (int age) throws WrongAgeException {  
 if (age < 0) {  
 throw new WrongAgeException ("Age cannot be negative");  
 }  
}

this.age = age ;

public int getAge () {  
 return age ;  
}

Class Son extends Father {

private int sonAge ;

public Son (int fatherAge, int sonAge) throws  
WrongAgeException {

super (fatherAge) ;

if (sonAge >= fatherAge) {

throw new WrongAgeException ("Son's age should  
be less than Father's age") ;

this.sonAge = sonAge ;

public int getSonAge () {

return sonAge ;

}

public class InheritanceExceptionHandling {

public static void main (String [] args) {

try {

int fatherAgeInput =

public class FatherSon {

public static void main (String [] args) {

Scanner sc = new Scanner (System.in) ;

System.out.println ("Enter father and son age") ;

int fa = sc.nextInt () ;

int sa = sc.nextInt () ;

try {

Son s = new Son (fa, sa) ;

System.out.println("Father's age : " + s.fatherAge);  
System.out.println("Son's age : " + s.sonAge);

catch (WrongAgeException e) {

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

Message (2) ;

Output —

Enter father's and son ages to add

51

20

Father's age : 51

Son's age : 20

Enter father and son age

40

41

Exception caught : Son's age should be less than  
father's age

## 7. Algorithm -

Step 1 : Start

Step 2 : Define WrongAgeException class that extends Exception

Step 3 : Define a class Father  
include private integer variable age

Step 4 : Implement a constructor that takes an age  
and check age if it is less than 0 and  
throw exception.

Step 5 : method to getAge

Step 6 : Define Son class that extends Father class

Step 7 : include private integer sonAge.

call super constructor with fatherAge

Step 8 : Checks if sonAge is equal to or greater  
than father age and throw exception.

Step 9 : In main method, get the input for son  
father Age.

Step 10 : Handle WrongAgeException by displaying  
the exception message.