1BM19CS150
SHIVANI GAHLOT

Page No.:
Date:
YOUVA

Lab Program - 8

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 & throws the exception NegativeAge() when the input age<0. In Son class, implement a constructor that takes both father & son's age and throws an exception WrongAge() if son's age is >= father's age.

```
import java.util.*;
class WrongAge extends Exception
{
  int f,s;
  WrongAge(int fAge, int sAge)
  {
        f = fAge;
        S = SAge;
  }
   public String toString()
  {
   return "Enter correct ages as Father's age can't
        be less than or equal to Son's age.";
  }
}
class NegativeAge extends Exception
{
  int xc;
  NegativeAge(int fAge)
  {  xc = fAge;
  }
```

```
    public String toString()
    {
      return "Age can't be negative.";
    }
}
class Father
{
  int fAge;
  Scanner in = new Scanner(System.in);
  Father() throws NegativeAge
  {
    System.out.println("Enter Father's Age:");
    fAge = in.nextInt();
    if (fAge<0)
    {
      throw new NegativeAge(fAge);
    }
  }
}
class Son extends Father
{
  int sAge;
  Scanner in = new Scanner(System.in);
  Son() throws NegativeAge, WrongAge
  {
    super();
    System.out.println("Enter Son's age:");
    sAge = in.nextInt();
    if (sAge<0)
    {
      throw new NegativityAge(sAge);
    }
    if (sAge>=fAge)
```

```
        {
            throw new WrongAge(fAge, sAge);
        }
    }
}
class exception_handling
{
public static void main(String args[])
    {
        try
        {
            Son s = new Son();
        }
        catch(NegativeAge n)
        {
        System.out.println("Exception: "+n);
        }
        catch(WrongAge w)
        {
        System.out.println("Exception: "+w);
        }
    }
}
```