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 and throws the excep tion Wrong Age( ) when the input age=father’s age.

```
import java.util.Scanner;
class fatherAgeException extends Exception
{
   public String toString()
   {
      return("Father's age is less than 0");
   }
}

class sonAgeException extends Exception
{
   int a;
   sonAgeException(int age)
   {
      a = age;
   }
   public String toString()
   {
      if(a<0)
       return("Son's age is less than 0");
```

```
        else
         return("Son's age is more than father's age");
    }
}
class Father
{
    public int agel;
    Scanner scan = new Scanner(System.in);
    int age1;
    Father()
    {
        System.out.println("Enter father's age: ");
        age1 = scan.nextInt();
    }
    void ex1() throws fatherAgeException
    {
        if (age1 < 0)
          throw new fatherAgeException();
    }
}
class Son extends Father
{
    public int age2;
    Son()
```

```
    {
        System.out.println("Enter son's age: ");
        age2 = scan.nextInt();
    }
    void ex2() throws sonAgeException
    {
        if(age2 < 0 || age2>super.age1)
        throw new sonAgeException (age2);
    }
}
class Main
{
    public static void main (String [] args){
        Son s = new Son();{
        try{
            s.ex1();
        }
        catch(fatherAgeException e)
        {
            System.out.println(e);
        }
        try
        {
            s.ex2();
```

```
        }
        catch(sonAgeException e)
        {
        System.out.println(e);
        }
        }

    }
}
```

Output:

```
Enter father's age:
12
Enter son's age:
45
Son's age is more than father's age
```