Name: Akram

USN: 1BM21CS013

Class: 3A

Lab 2 Question: Write a program that demonstrates the handling of exceptions in an inheritance tree. Create a base class called “Father” and a derived class called “Son” which extends the base class. In the father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age<0. In the son class, implement a constructor that cases both father and son’s age and throws an exception if the son’s age is >= father’s age.

Program:

30/12/22

Exception Handling

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 cases 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
{
    String msg = new String();
    WrongAge()
    {
        msg = "Age Error";
    }
    WrongAge(String ss)
    {
        msg = ss;
    }
    public String toString()
    {
        return msg;
    }
}

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

```
class Father extends InputScanner {

    int fatherAge;
    Father() throws WrongAge
    {
        Scanner ss = new Scanner(System.in);
        System.out.println("Enter father's age : ");
        fatherAge = s.nextInt();
        if (fatherAge<=0)
            throw new WrongAge("Age cannot be 0 or negative");
    }
    void display()
    {
        System.out.println("Father's age : " + fatherAge);
    }
}

class Son extends father {
    int sonAge;
    Son() throws WrongAge {
        System.out.println("Enter son's age: ");
        sonAge = s.nextInt();
        if (sonAge >= fatherAge)
            throw new WrongAge("Son's age cannot be greater than father's age");
        else if (sonAge<=0)
            throw new WrongAge("Age cannot be 0 or negative");
    }
    void display()
    {
        super.display();
        System.out.println("Son's age : "+sonAge);
    }
}
```

```
class Main {
    public static void main (String args[])
    {
        try
        {
            Son S1 = new Son();
            S1.display();
        }
        catch (WrongAge wa)
        {
            System.out.println(wa);
        }
    }
}
```

Output:

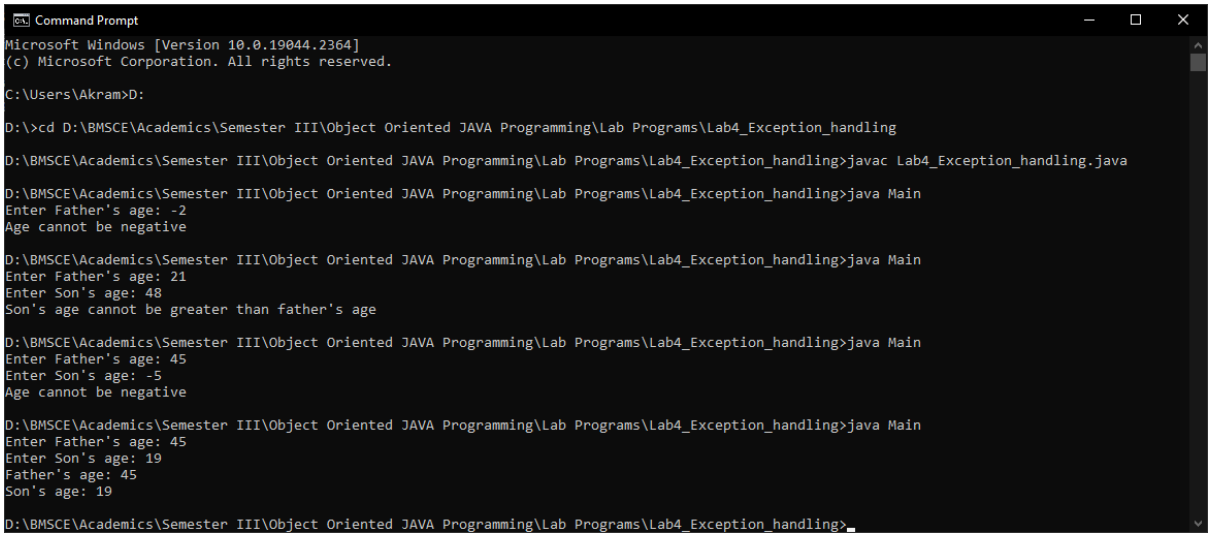
Enter father's age: -2
Age cannot be 0 or negative

Enter father's age: 21
Enter Son's age: 48
Son's age cannot be greater than father's age

Enter father's age: 45
Enter son's age: 19
Father's age: 45
Son's age: 19

30/12/22

Output:


Command Prompt
Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Akram>D:
D:\>cd D:\BMSCE\Academics\Semester III\Object Oriented JAVA Programming\Lab Programs\Lab4_Exception_handling
D:\BMSCE\Academics\Semester III\Object Oriented JAVA Programming\Lab Programs\Lab4_Exception_handling>javac Lab4_Exception_handling.java
D:\BMSCE\Academics\Semester III\Object Oriented JAVA Programming\Lab Programs\Lab4_Exception_handling>java Main
Enter Father's age: -2
Age cannot be negative
D:\BMSCE\Academics\Semester III\Object Oriented JAVA Programming\Lab Programs\Lab4_Exception_handling>java Main
Enter Father's age: 21
Enter Son's age: 48
Son's age cannot be greater than father's age
D:\BMSCE\Academics\Semester III\Object Oriented JAVA Programming\Lab Programs\Lab4_Exception_handling>java Main
Enter Father's age: 45
Enter Son's age: -5
Age cannot be negative
D:\BMSCE\Academics\Semester III\Object Oriented JAVA Programming\Lab Programs\Lab4_Exception_handling>java Main
Enter Father's age: 45
Enter Son's age: 19
Father's age: 45
Son's age: 19
D:\BMSCE\Academics\Semester III\Object Oriented JAVA Programming\Lab Programs\Lab4_Exception_handling>