

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 exception Wrong Age() when the input age=father’s age.

CODE:

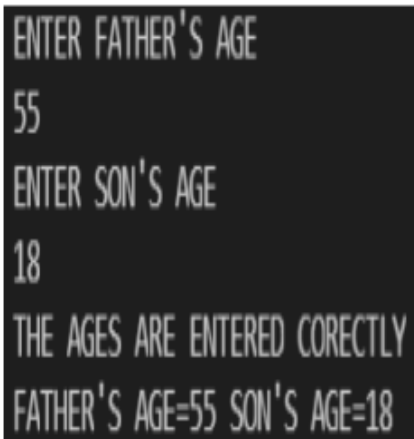
```
import java.util.*;
import java.lang.String;
import java.awt.*;
import java.awt.event.*;
class WrongAge extends Exception{
    int age;
    WrongAge(int x)
    {
        age=x;
    }
    public String toString()
    {
        return "AGE OF SON="+age+" IS ENTERED
INCORRECTLY";
    }
}
```

```
}  
}  
class father  
{  
int a;  
father(int x)  
{  
a=x;  
}  
}  
class son extends father{  
int age;  
son(int fage,int sage){  
super(fage);  
age=sage;  
}  
void compute() throws WrongAge{  
if(age>=a)  
{  
throw new WrongAge(age);  
}  
}
```

```
else{
    System.out.println("THE AGES ARE ENTERED
CORECTLY");
    System.out.println("FATHER'S
AGE="+a+"\t"+"SON'S
AGE="+age);
}
}
}
class expmain
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("ENTER FATHER'S AGE");
        int f=s.nextInt();
        System.out.println("ENTER SON'S AGE");
        int so=s.nextInt();
        son ss=new son(f,so);
        try{
            ss.compute();
```

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

OUTPUT:

A screenshot of a terminal window with a dark background and light green text. The text shows the program's execution flow: it prompts for the father's age, receives 55, prompts for the son's age, receives 18, and then prints a confirmation message and the entered ages.

```
ENTER FATHER'S AGE
55
ENTER SON'S AGE
18
THE AGES ARE ENTERED CORECTLY
FATHER'S AGE=55 SON'S AGE=18
```