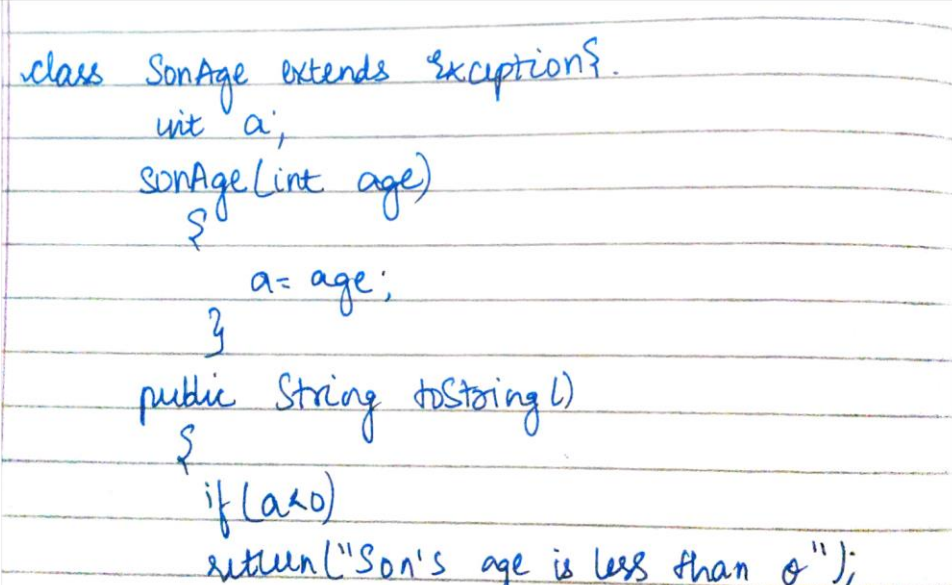


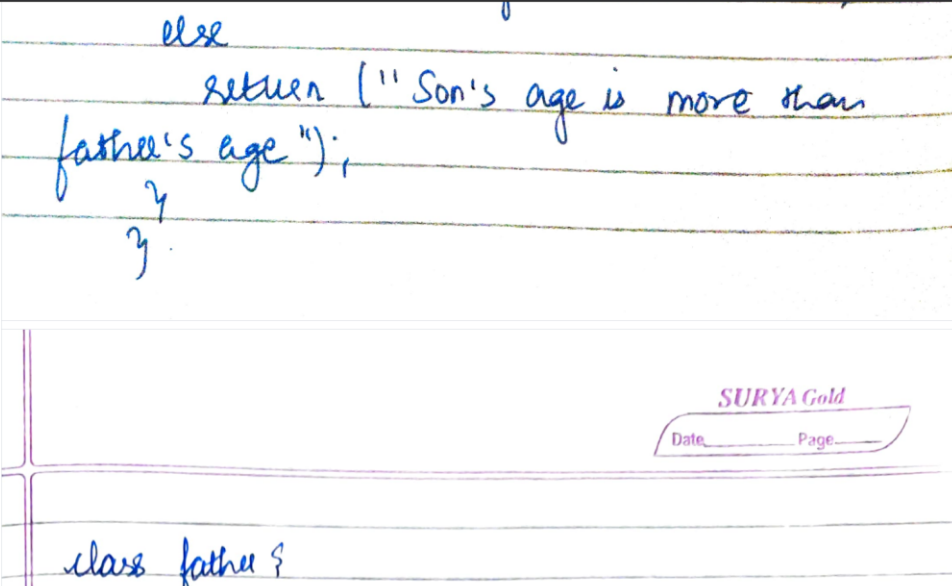
LAB PROGRAM 8

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 and throws the exception Wrong Age() when the input age=father's age.



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

The image shows a handwritten Java code snippet for a class named `SonAge` which extends `Exception`. The code includes a constructor `SonAge(int age)` that assigns the parameter `age` to the instance variable `a`. It also has a `toString()` method that checks if `a` is less than 0 and returns a string message if true. The code is written on lined paper and is part of a video recording, as evidenced by the browser tabs and taskbar at the bottom.



```
else  
    return ("Son's age is more than  
father's age");  
}  
  
class Father {  
    int f;  
    Father(int age)  
    {  
        f = age;  
    }  
}
```

The image shows the continuation of the handwritten Java code. It includes an `else` block for the `toString()` method of `SonAge`, which returns a message if the son's age is more than the father's age. Below this, the `Father` class is defined with an instance variable `f` and a constructor `Father(int age)` that assigns `age` to `f`. The code is written on lined paper and is part of a video recording, as evidenced by the browser tabs and taskbar at the bottom.

```
public int age1;  
Scanner ss= new Scanner(System.in);  
father() {  
    SOP("Enter father's age:");  
    age1 = ss.nextInt();  
}  
void exception1() throws fatherAge {  
    if (age1 < 0)  
        throw new fatherAge();  
}
```

```
class son extends father {  
    public int age1;  
    son() {  
        SOP("Enter son's age:");  
        age2 = ss.nextInt();  
    }  
    void exception2() throws sonAge {  
        if (age2 < 0 || age2 > super.age1)  
            throw new sonAge(age2);  
    }  
}
```

```
class fatherSon
{
    public static void main(String args[])
    {
        son s = new son();
        try
        {
            s.exception1();
        }
        catch (fatherAge e)
        {
            end1();
        }
    }
}
```

```
} } SDP(e); }
try {
    s.exception2();
}
catch (sonAge e)
{
    SDP(e);
}
}
}
```

OUTPUT:

Week 10- Lab Program 7 and 8 x OOI-Lab-Programs/lab8_output... x New Tab x +

github.com/Nayanaj117/OOI-Lab-Programs/blob/master/WEEK%2010/lab8_output.pdf

master OOI-Lab-Programs / WEEK 10 / lab8_output.pdf Go to file ...

Nayanaj117 Add files via upload Latest commit bae05cd on 24 Nov History

1 contributor

196 KB Download

```
C:\Users\91966\Desktop\OOJLAB>javac fatherson.java
C:\Users\91966\Desktop\OOJLAB>java fatherson
ENTER FATHER'S AGE
50
ENTER SON'S AGE
50
AGE OF SON=50 IS WRONG
```

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

Type here to search

21:13 28-12-2020