Lab 7 30/1/24

→ write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called Father and derived class called Son which extends 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 & son's age and throws an Exception if son's age is >= father's age.

```
import java.util.Scanner;
class WrongAge extends Exception{
public WrongAge(String A){
    super(A);
}}
class Father{
int fatherAge;
Scanner sc = new Scanner(System.in);
Public void validAge() throws WrongAge{
System.out.println("Enter fathers age");
fathers Age = sc.nextInt();
if(fatherAge <=0){
    throw new WrongAge("Invalid father's age");
}}
}
Class Son extends Father{
int sonAge;
Scanner sc = new Scanner(System.in);
public void validAge() throws WrongAge{
    System.out.println("Enter sons Age");
    sonAge = sc.nextInt();
    super.validage();
```

```
if (sonAge >= fatherAge){
    throw new WrongAge("Sons age can't be
    greater than father's age");
  }
}
else if (sonAge <0){
    throw new WrongAge ("Invalid son age
  }
}}

public class MyMain{
  public static void main(String[] args){
    System.out.println("Sanskriti [illegible] 2023[illegible]
    Son obj = new Son();
      try {
        obj.validAge();
      }
    catch (WrongAge e){
    System.out.println ("Exception "+e.getMessage
      }}}
```

Output Sanskriti [illegible]

Enter Sons age

30

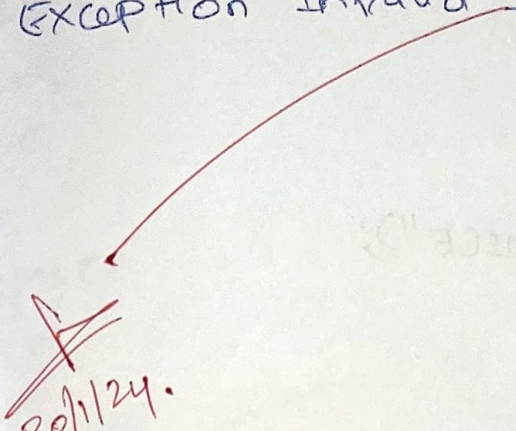
Enter father's age

22

Exception Sons age cant be greater than Father's

age

Enter Sons age
→
Enter Fathers age
20
Exception Invalid son age.



30/1/24.

(1);