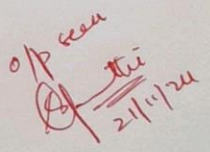
21/11/24

LAB-7

Q. WAP that demonstrates handling of exceptions in inheritance tree. Create a base class father and derived class son which extends base class. In Father class, implement a constructor which takes the age and throws exception WrongAge() when the input age<0. In son class, implement a constructor that uses both father and sons' ~~class~~ age and throws an exception if sons' age >= fathers' age.

```
class WrongAge extends Exception{
    String message;
    WrongAge(String message){
        this.message=message;
    }
    public String toString(){
        return "WrongAge Exception: "+message;
    }
}
class Father{
    int fAge;
    Father(int age) throws WrongAge{
        if(age<0){
            throw new WrongAge("Fathers' age can not be negative!");
        }
        fAge=age;
    }
}
class Son extends Father{
    int sAge;
    Son(int fAge, int sAge)throws WrongAge{
        super(fAge);
        if (sAge<0){
            throw new WrongAge("Sons' age can not be negative!");
        }
        if(sAge>=fAge){
            throw new WrongAge("Sons' age can not be greater than or
                                equal to Fathers' age!");
    }
```

```
        this.sAge = sAge;
    }
}
public class Main{
    psvm (String[] args){
      try{
          Father father1=new Father(40);
          Son son1=new Son(40,20);
          S.O.P("Fathers' age: "+father1.fAge+", Sons' Age: "+son1.sAge);
          Father father2=new Father(-5);
      }
      catch(WrongAge e){
          S.O.P(e);
      }
      try{
          Son son2=new Son(35,40);
      }
      catch(WrongAge e){
          S.O.P(e);
      }
      try{
          Son son3=new Son(50,-10);
      }
      catch(WrongAge e){
          S.O.P(e);
      }
    }
}
```



output:

Fathers' Age: 40, Sons' Age: 20

WrongAge Exception: Fathers' Age can not be negative!

WrongAge Exception: Sons' Age can not be greater than or equal to Fathers' Age!

WrongAge Exception: Sons' Age can not be negative!

```
class WrongAge extends Exception {
	String message;
	WrongAge(String message) {
		this.message = message;
	}
	public String toString() {
		return "Wrong Age Exception: " + message;
	}
}

class Father {
	int fAge;
	Father(int age) throws WrongAge {
		if (age < 0) {
			throw new WrongAge("Father's age cannot be negative!");
		}
	fAge = age;
	}
}

class Son extends Father {
	int sAge;
	Son(int fAge, int sAge) throws WrongAge {
		super(fAge);
		if(sAge>0){
			throw new WrongAge("Sons' age can not be negative!");
		}
		if (sAge > =fAge) {
			throw new WrongAge("Son's age cannot be greater than or equal to father's
		age!");
		}
```

```
                this.sAge = sAge;
        }
}

public class Main {
        public static void main(String[] args) {
                try {
                        Father father1 = new Father(40);
                        Son son1 = new Son(40, 20);
                        System.out.println("Fathers' age:"+father1.fAge+",Sons' Age:"+son1.sAge);
                        Father father2 = new Father(-5);
                }
                catch (WrongAge e) {
                System.out.println(e);
                }
                try{
                        Son son2 = new son(35, 40);
                }
                catch(WrongAge e) {
                        System.out.prinltn(e);
                }
                try{
                        Son son3 = new son(50, -10);
                }
                catch(WrongAge e){
                        System.out.prinltn(e);
                }
        }
}
```


C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.
C:\317\7_Exception>javac Main.java
C:\317\7_Exception>java Main
Father's age: 40, Son's age: 20
WrongAge Exception: Father's age cannot be negative!
WrongAge Exception: Son's age cannot be greater than or equal to Father's age!
WrongAge Exception: Son's age cannot be negative!
C:\317\7_Exception>