

WEEK 7 :

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 uses both father and son’s age and throws an exception if son’s age is >=father’s age.

Source Code :

```
import java.util.Scanner;

class WrongAgeException extends Exception {    public
WrongAgeException(String message) {        super(message);
    }
} class SonAgeException extends Exception {    public
SonAgeException(String message) {        super(message);
    }
} class Father {    int age;    public Father(int age) throws
WrongAgeException {        if (age <= 0) {            throw new
WrongAgeException("Wrong age");
        }
        this.age = age;
    }    public int getAge() {
return age;
    }
} class Son extends Father {    int
sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
super(fatherAge);        if (sonAge >= fatherAge) {
```

```

        throw new SonAgeException("Son's age cannot be greater than or equal to father's age");
    }
    if(sonAge <= 0){
        throw new
WrongAgeException("Wrong age");
    }
    this.sonAge = sonAge;
}
public int getSonAge() {
return sonAge;
}
}

public class FatherSon{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter
Father's Age: ");
        int fatherAge = sc.nextInt();
        System.out.print("Enter Son's Age: ");
        int sonAge =
sc.nextInt();
        try {
            Son son = new Son(fatherAge, sonAge);
            System.out.println("Accepted Succesfully");
        }
        catch (WrongAgeException e) {
            System.out.println(e.getMessage());
        }
        catch (SonAgeException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

Output :

```
Enter Son's Age: 26
Accepted Successfully
PS C:\Users\satis\OneDrive\Documents\ooj_lab> javac FatherSon.
PS C:\Users\satis\OneDrive\Documents\ooj_lab> java FatherSon
Enter Father's Age: 30
Enter Son's Age: 32
Son's age cannot be greater than or equal to father's age
```

```
Enter Father's Age: 30
Enter Son's Age: 0
Wrong age
```

OBSERVATION:

Prgm 7

21/11/24

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

```
class WrongAgeException extends Exception {  
    public WrongAgeException(String message) {  
        super(message);  
    }  
}  
  
class Father {  
    protected int age;  
    public Father(int age) throws WrongAgeException {  
        if (age < 0) {  
            throw new WrongAgeException("Father's age  
            cannot be negative");  
        }  
        this.age = age;  
    }  
}  
  
class Son extends Father {  
    private int sonAge;  
  
    public Son(int fatherAge, int sonAge) throws WrongAgeException {  
        super(fatherAge);  
        this.sonAge = sonAge;  
        if (sonAge >= fatherAge) {  
            throw new WrongAgeException("Son's age  
            cannot be greater than or equal to father's age");  
        }  
    }  
}
```

```

super(fatherAge);
if (sonAge < 0) {
    throw new WrongAgeException("Son's age
    cannot be negative");
}
if (sonAge >= fatherAge) {
    throw new WrongAgeException("Son's age
    cannot be greater than or equal to
    father's age");
}

```

```

    this.sonAge = sonAge;
}

```

```

}

```

```

public String toString() {

```

```

    return "Father's Age : " + age + "Son's Age : " + sonAge;
}

```

```

}

```

```

public class ExceptionInheritance Demo {
    public static void main(String [] args) {
        try {

```

```

            Father father = new Father(45);

```

```

            System.out.println("Father is created with
            age : " + father.age);

```

```

            Son son = new Son(45, 20);

```

```

            System.out.println(son);

```

```

        } catch (WrongAgeException e) {

```

```

            System.out.println("Exception occurred" +
            e.getMessage());

```

```

        }
    }
}

```



```

try {
    Son invalidSon = newSon(40, 40);
} catch (WrongAgeException e) {
    System.out.println("Exception occurred" + e.getMessage());
}

try {
    Father invalidFather = newFather(-5);
} catch (WrongAgeException e) {
    System.out.println("Exception occurred" + e.getMessage());
}
}
}

```

Father created with age : 45

Father's Age : 45, Son's Age : 20

Exception occurred: Son's age cannot be greater than or equal to father's age.

Exception occurred: Father's age cannot be negative.

28/11/24