Lab - 7

Write a program that demonstrates handling of exception in inheritance. 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 & throws the exception wrongAge() when the input age < 0. In son class, implement a constructor that takes both father & son's age & 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 Exception {
        System.out.println("Enter father's age:");
        fatherAge = sc.nextInt();
        if (fatherAge <= 0) {
            throw new WrongAge("Invalid fathers Age, Age cannot be negative");
        }
    }
}
```

```
class son extends father{
    int sonAge;
    Scanner sc=new Scanner(System.in);
    public void validAge() throws
    wrongAge{
        System.out.println("Enter son's
        age :");
        sonAge=sc.nextInt();
        super.validAge();
        if(sonAge>=fatherAge){
            throw new wrongAge("sons age
    cant be greater then fathers age");
    }
    Else if(fatherAge==sonage){
    Throw new wrongage
    System.out.println("fathers and son's
        age cannot be the same");
    }
    Else if(sonAge<0){
        throw new wrongAge("Invalid
      son age");
    }
}
}

Public class Mymain{
    public static void main(string[] args
    { Son obj=new son();
        try{
            Obj.validAge();
        }
        catch(wrongAge e){
            System.Out.println("exception"+
```

```
            e.getMessage());
        }
    }
}
```

output: Enter sons age
20
Enter father's age: 24
Exception sons age & father age
cannot be same

30.01.24