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=father's age.

```
package DemoExceptions;

public class WrongAgeException extends Exception {
   public WrongAgeException(String message) {
      super(message);
   }
}

}
```

```
package DemoExceptions;

public class Father {
        int age;

   public Father(int age) throws WrongAgeException {
      if (age < 0) {
         throw new WrongAgeException("Age cannot be negative");
      }
      this.age = age;
   }


}
```
```
package DemoExceptions;

public class Son extends Father {
        int sonAge;

        public Son(int fatherAge, int sonAge) throws WrongAgeException {
           super(fatherAge);
           if (sonAge >= fatherAge) {
              throw new WrongAgeException("Son's age cannot be greater than or equal to Father's age");
           }
           this.sonAge = sonAge;

}
}
```
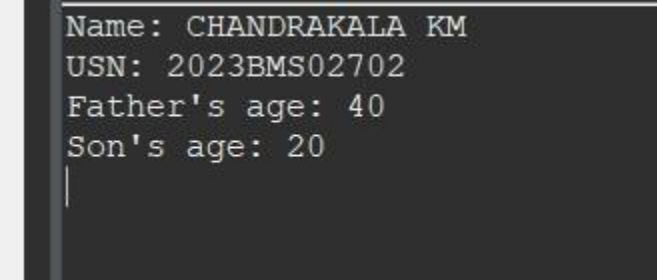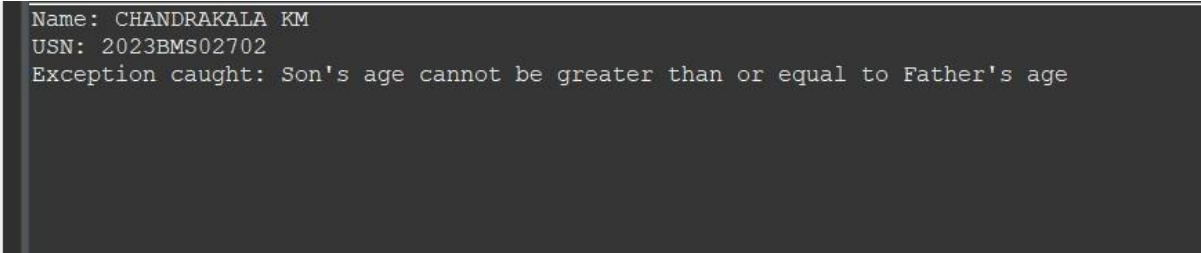```
package DemoExceptions;

public class Excepmain {
   public static void main(String[] args) {
        System.out.println("Name: CHANDRAKALA KM");
              System.out.println("USN: 2023BMS02702");
```

```
        try {
            int fatherAge = 40;
            int sonAge = 20;
            Son son = new Son(fatherAge, sonAge);
            System.out.println("Father's age: " + fatherAge);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

OUTPUT:

```
Name: CHANDRAKALA KM
USN: 2023BMS02702
Father's age: 40
Son's age: 20
```

```
Name: CHANDRAKALA KM
USN: 2023BMS02702
Exception caught: Son's age cannot be greater than or equal to Father's age
```

1) To write a program that demonstrates handling of exception 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 & throws the exception wrongAge() when the input ag <0. In Son class, implement a Constructor that takes both father & son's age and throws an exception if son's age is >= father's age.

```java
Class wrongAgeException extends Exception {
    public wrongAgeException (String message) {
        super (message);
    }
}

Class Father {
    int age;
    public Father (int age) throws wrongException {
        if (age < 0) {
            throw new wrongAgeException ("Age cannot be negative");
        }
        this.age = age; }
}

Class Son extends Father {
    int sonAge;
    public Son (int fatherAge, int sonAge) throws
wrongAgeException {
    super (fatherAge);
    if (sonAge >= fatherAge) {
        throw new wrongException ("son's age cannot be
greater than our equal to Father's age"); }
    this.sonAge = sonAge; }
}
```

```java
public class main123 {
    public static void main (String [] args) {
        try {
            int fatherAge = 10;
            int sonAge = 20;
            Son son = new Son (fatherAge, SonAge);
            System.out.println ("Father's age: " + fatherAge);
            System.out.println ("Son's age: " + son.sonAge);
        }
        Catch (wrongAge Exception e) {
            SOP ("Exception Caught:" + e.getmessage()); }
    }
}
```

output:

Exception Caught: son's age cannot be greater than
or equal to Father's age.

Father's age: 10
son's age: 20