7) 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 and throws the Exception WrongAge() when the input age<0 in Son ~~marks~~ class. ; implement a constructor that uses both father and son's age and throw an Exception if son age >= father Age.

```
-> class WrongAge extends Exception {
       WrongAge (String message) {
           super(message);
```

```
}.
class father { int Age;
   father (int age) {          → throws wrongAge
      if (age < 0) {
      throw new wrongAge ("Age cannot be Negative")
      }                                                ;
     this.age = age;
   }
 }                              throws wrongAge
class son extends father (~~int age1, int age2~~) {
      son (int fage, int sage) {
        super (fage);
        if (sage >= fage) {
        throw new ~~exception~~ wrongAge ("
             son age cant greater than fatherage");
        }
        this.fage = fage;
        this.sage = sage;
      }
 }.
class main {
   psvm (String Args[]) {
     for (;;) {
    Scanner sc = new Scanner (System.in);
    try {
     int fage;
     int sage;
     System.o.p ("Enter father age");
     fage = sc.nextInt ();
     S.o.p (" enter son age:")
     sage = sc.nextInt ();
Son s = new Son (fage, sage);
}
 catch (WrongAge wa) {
   System.out.println(wa);
  }
 }
}
}
```

```
op :  Enter father's age :-1
      Enter son's age = 1
      Error : Age cannot be negative
      Enter father's age : 20
      Enter son's age : 21
      Error! : Son's age > father age
```

[illegible] which creates two threads