Program 7

Q] 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.

CODE:

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

Program 7

```
public static void main(String[] args) {
        try {
            Father father = new Father(45);
            Son son = new Son(45, 20);
            System.out.println("Father's age: " + father.age);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
        try {
            Son invalidSon = new Son(30, 35);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
        try {
            Father invalidFather = new Father(-5);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
   }
}
```

OUTPUT:

```
Father's age: 45
Son's age: 20
Exception: Son's age cannot be greater than or equal to Father's age
Exception: Father's age cannot be negative
```

OBSERVATION:

Program 7

Del kosite a program that demonstrates handling of oceptions in inhustance tree. create as base class called "Father" and derived class called "Son" which actends the base class. called "Son" which actends the base class. In Father which outside the base class. In Father which outside the base class. In Father which outside the class, implement a constructor which takes he class, implement a constructor which takes he class, implement a constructor that uses booth father and son's age and throws an acception wrong Age () when the input age 20. In son boring Age () when the input age 20. In son boring Age () when the input age 20. In son the son's age and throws an tothe father and son's age and throws an techtion of father and son's age and throws an ecception of father and son's age and throws an ecception of father and son's age and throws an ecception of father and son's age and throws an ecception of father and son's age and throws an ecception of father and son's age and throws an ecception of father and son's age and throws an ecception of father and son's age and throws an ecception of father and son's age is a father age.

Code:

class wrong age extends exception (
public wrong age (Strong massage) (
Super (Missage);

9

3

class Father ?

int father Age;

techlic Father (I'nt age) throws wrong Age (
if (age < 0) {
throw new wrong Age ("Father's age
cannot be neg able.");

```
this. father age = age;
 class Sen octindo Father ?
          public son (Pat fatherage, and sonage)
         int son ages
          throws wrong age! galades
                if (Son Age > = fatherAge) {
               throw new wronglys ("Son's age can't be
         quater or equals fither Age");
        this . Son Age = Son Age ;
 3
public class Main 1
       PSVM() 1.
      try 1
         Son son3 = new Son (-1, 10);
         son son = new son (50,25);
          sout (" Fathers age: " + son! fatherage
          + " son's Age ! " + son 1. son Age);
      3
     · catch (wrong Age &) {
           Sout (" Exception caugh: ">+ E. get MENS-eg oct);
      3
3
        Hooutput
```

Program 7

Son Age can't be negative

Son Age can't be represent than fathor's age.

The solution of the