LAB PROGRAM 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.

CODE:

```
import java.util.Scanner;
class WrongAge extends Exception {
    WrongAge(String msg) {
        super(msg);
    }
}
class Father {
    int age;
    Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Age cannot be negative");
        this.age = age;
    }
class Son extends Father {
    int sonAge;
    Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age should be less than Father's age");
        this.sonAge = sonAge;
    }
}
public class Exceps {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Chaitanya N");
        System.out.println("1BM22CS076");
        System.out.println();
        try {
            System.out.print("Enter father's age: ");
```

```
int fatherAge = scanner.nextInt();
    System.out.print("Enter son's age: ");
    int sonAge = scanner.nextInt();

    Son son = new Son(fatherAge, sonAge);
    System.out.println("Father's age: " + fatherAge);
    System.out.println("Son's age: " + sonAge);
} catch (WrongAge e) {
    System.out.println("Error: " + e.getMessage());
} catch (Exception e) {
    System.out.println("Error: Invalid input");
} finally {
    scanner.close();
}
}
```

OUTPUT:

```
PS C:\Users\ADMIN\Documents\CSE III\java prgms\ cd "c:\Users\ADMIN\Documents\CSE III\java prgms\"; if ($?) { javac Exceps.java }; if ($?) { java Exceps }
Chaitanya N
1BM22CS076
Enter father's age: -5
Enter son's age: -2
Error: Age cannot be negative
PS C:\Users\ADMIN\Documents\CSE III\java prgms\ cd "c:\Users\ADMIN\Documents\CSE III\java prgms\"; if ($?) { javac Exceps.java }; if ($?) { java Exceps.java };
Chaitanya N
1BM22CS076
Enter father's age: 50
Enter son's age: 85
Error: Son's age should be less than Father's age
PS C:\Users\ADMIN\Documents\CSE III\java prgms> cd "c:\Users\ADMIN\Documents\CSE III\java prgms\" ; if ($?) { javac Exceps.java } ; if ($?) { java Exceps }
Chaitanya N
1BM22CS076
Enter father's age: 40
Enter son's age: 20
Father's age: 40
Son's age: 20
PS C:\Users\ADMIN\Documents\CSE III\java prgms>
```

6) Novele perogenem that demonstrates handlenos of exceptions on anherstowns love . Coverede bare relove "Father" and desput was ebulled "Son" which extends the bore relove. In Father dose, amplement construction who in tokus the age & therone exception would age < 0. In son closes, amplement of construction that were both father and son's age & therein an exception 26 son's age & so > = father's age

```
-> 9mport forma. util. Scanner;

ulars Wowng Age extends Exceptions

{
Wowng Age (Storng msq))

$ suber (msq);

3

class Forther
```

class Forther

Part age;

Father (ent age) throws Wrong Age

Pop (oug < < 0)

therew new Weioner Age (" Age cannot be negative")

thes. orge = orge's

3

illows Son extends Father

§ ant son Age;

8on (ant Jother Age, ant son Age) therewas Wowing Agel)

super (fatherAge);

Pb (son Aqe > = fatherdore) {

thruon none Woung Agel" Son age connot be guater thour forther orge") 5

y & thes. son Age = son Age;

```
public clark Horn
    public stodie vord moun (strong ougs (J)
              SC = new Sconney (System. 9n);
     Swanney
   tony
        System. out. perentin (a enter forther songe: ");
        ent fatherage = sc. nextont ();
        ant conduce = su. number ();
        Father f = now i=outher (fortherdge);
        System. out. perently (" Fother's oug e:" + fathering );
        Son s = neve son (forthertoge, son dge);
        System. out puint in ("Fother's orge: " + fortherdge);
        System. out. perontin (" Son's orga: " + son Ag 1);
      cout in ( Warong Age e)
           System. out perentin ( 0);
       colin (Except pon e)
              System. out buendin (" from e);
       banoully &
                 sudose ();
Output:
   Enter son's ouge; so
```

1) Enter father : a age: -8 Age connot be negatione

2) Enter forther's orge: 210 Enter son's orge: Father 's onge: 40 Son's orge should be less thom Fother age.