## **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.

```
1. write a program that demonstrates handling of exceptions in inheritance trice. Create a base class called "tather" and
derived class called "son" which extends the base class. In
Father class, implement a construction which takes the age and therows the exception wrongfge () when the input
age < 0. In son class, implement a constructor that
uses both father and son's age and thrown an exception
if son's age is >= fatheris age.
class whong Age extends Exception (
Public Warong Age (storing message) [
Super (message);
                       friends ("From: "+ 0-93
class Father
    fublic Father (int age) throws WarangAge (
i) (age < 0) [
thorow new warangAge ("Age Cannot be negative for
Father");
        System out println ("Father's age is" + this age);
                               aried be negative for to
   Jublic 36n (int fatherage, int sonage) therows waringedge
     if (sontage > = fatherage) {
      throw new Wenergage ("son's age rannot be greater than on
```

System out println ("son's age is "+ this age 10 1 10 0 1 2 Wal. Father fatheril = new Father (-5); catch (wormange e) System-out println ("Ervion: tony 1 , Son won1 = new son (40, \$5); catch (WoungAge e) System out println ("From: Son Gon2 = new Son (40, 30); I catch (whongAge e) [ System. out. println ("Eturos: "te get Messagel Ensign: Age Cannot be negative for Father Ensign: son's age cannot be greater than one be greater than on equal Father's age is 4011 Son's age not reterred Jennes ope inci see sinesthat et leaper -

```
class WrongAge extends Exception {
  public WrongAge(String message) {
    super(message);
  }
}
class Father {
  int age;
  public Father(int age) throws WrongAge {
    if (age < 0) {
       throw new WrongAge("Age cannot be negative for Father.");
     }
    this.age = age;
    System.out.println("Father's age is " + this.age);
  }
}
class Son extends Father {
  int sonAge;
  public Son(int fatherAge, int sonAge) throws WrongAge {
    super(fatherAge);
```

```
if (sonAge >= fatherAge) {
       throw new WrongAge("Son's age cannot be greater than or equal to Father's age.");
     }
     this.sonAge = sonAge;
     System.out.println("Son's age is " + this.sonAge);
  }
}
public class Age {
  public static void main(String[] args) {
     try {
       Father father1 = new Father(-5);
     } catch (WrongAge e) {
       System.out.println("Error: " + e.getMessage());
     }
     try {
       Son son1 = new Son(40, 45);
     } catch (WrongAge e) {
       System.out.println("Error: " + e.getMessage());
     }
     try {
       Son son2 = \text{new Son}(40, 30);
     } catch (WrongAge e) {
       System.out.println("Error: " + e.getMessage());
     }
  }
```

}

## D:\24BMSCE>javac Age.java

D:\24BMSCE>java Age

Error: Age cannot be negative for Father.

Father's age is 40

Error: Son's age cannot be greater than or equal to Father's age.

Father's age is 40

Son's age is 30