WEEK_6

QUESTION:

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 cases both father and son’s age and throws an exception if son’s age is >=father’s age.

7] Write a program that demonstrate 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.

~~class Son extends Father{~~
~~ftm~~

```
import java.util.Scanner;
class WrongageException extends Exception{
        public String toString(){
            return ("Negative age can't be accepted");
    }
  }
class   AgeException extends Exception{
            public String toString(){
            return ("son can't be older than
                  father");
        }
  }
class  Father{
        int father_age;
        Father (int x) throws WrongageException{
            father_age = x;
            if (father_age <0){
                throw new WrongAgeException()
            }
  } }
```

```
class son extends Father{
        int son-age;
        Son( int x, int y) throws AgeException,
        wrongAgeException{
            super(x);
            son-age = y;
            if (son-age<0){
                throw new wrongAgeException();
            }
         if (son-age >= father-age){
                throw new AgeException();
         }
        } }

class Lab-7 {
       public static void main (String[] args){
         try{
            Scanner s = new Scanner (System.in);
            System.out.println("Enter father's and
                               son's ages");
            int x = s.nextInt();
            int y = s.nextInt();
            Son so= new Son (x, y);
            System.out.printf ("Father is %d years
            old and son is %d years old", so.father
            age, so.son_age);
         }
         catch (AgeException a){
              System.out.println(a);
         }
         catch (wrongAgeException wa){
                 System.out.println(wa);
              }
         catch (Exception e){
```

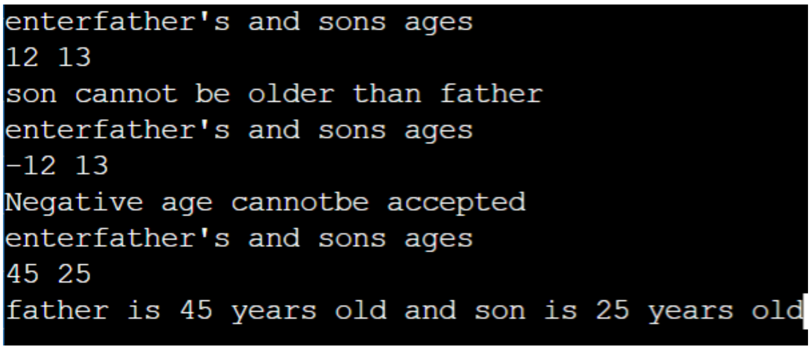
```
        System.out.println ("Enter valid values");
      }
    }
  }
}
```

Output :

```
Enter father's and son's ages
12    13
Son can't be older than father

Enter father's and son's age
-12    13
Negative age can't be accepted.

Enter father's and son's age
12    -13
Negative age can't be accepted.

Enter father's and son's age
45    25
Father is 45 years old and son is 25
years old.
```

20/12/22

OUTPUT:



```
enterfather's and sons ages
12 13
son cannot be older than father
enterfather's and sons ages
-12 13
Negative age cannotbe accepted
enterfather's and sons ages
45 25
father is 45 years old and son is 25 years old
```