PROGRAM 6

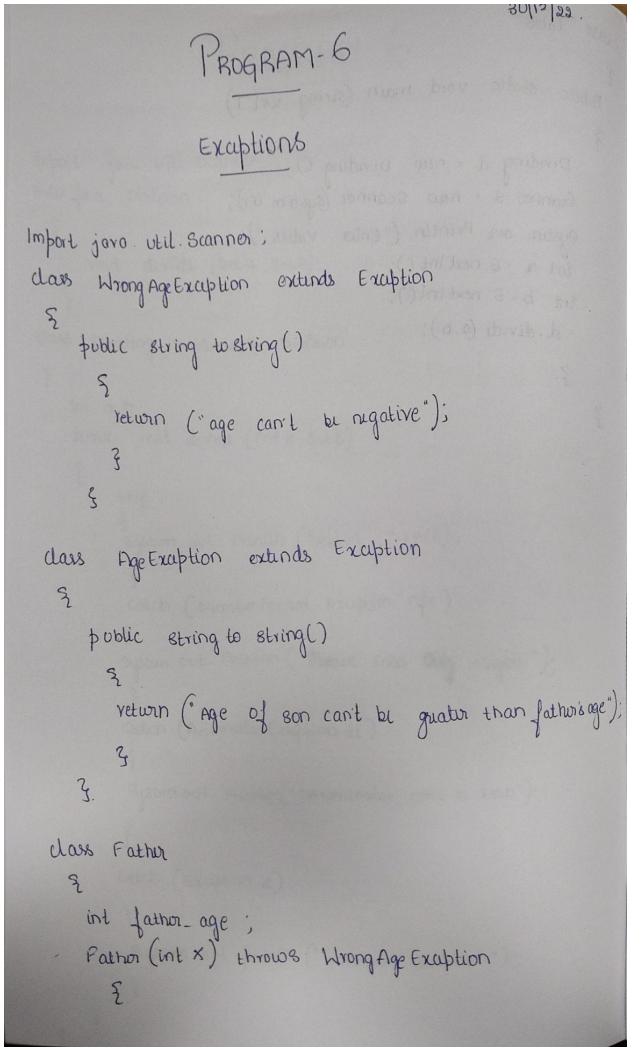
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

```
import java.util.Scanner;
class WrongAgeException extends Exception{
        public String toString(){
                return ("age can't be negative");
        }
}

class AgeException extends Exception{
        public String toString(){
                return("Age of son can't be greater than father's age");
        }
}

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 lab7
{
        public static void main(String args[]) {
                try {
                        Scanner s=new Scanner(System.in);
                        System.out.println("Enter father's  age :");
                        int x=s.nextInt();
                        System.out.println("Enter son's age:");
                        int y=s.nextInt();
                        Son S=new Son(x,y);
                        System.out.println("Father age is " + S.father_age +
"\nSon age is " + S.son_age);
        }
                catch (WrongAgeException wa) {
                        System.out.println(wa);
                }
                catch (AgeException a){
                        System.out.println(a);
                }
```

```
                catch (Exception e){
                        System.out.println("enter valid numbers");
        }
    }
}
```

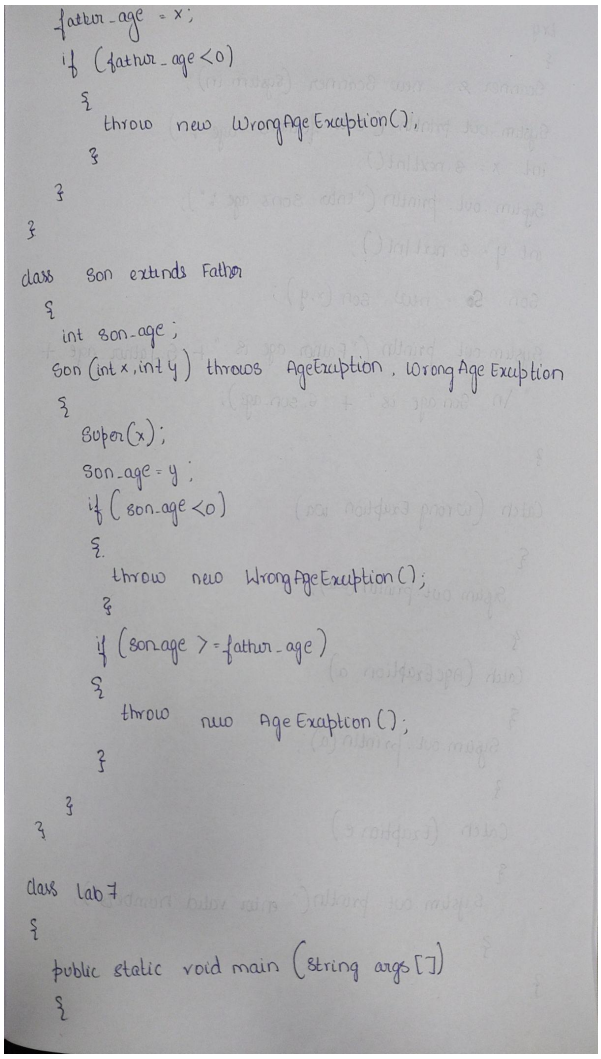
observation:



PROGRAM-6

Exceptions

```
Import java.util.Scanner;
class WrongAgeException extends Exception
 {
   public string toString()
    {
      return ("age can't be negative");
    }
 }

class AgeException extends Exception
 {
   public string to string()
    {
      return ("Age of son can't be grater than father's age");
    }
 }

class Father
 {
   int father_age ;
   Father (int x) throws WrongAge Exception
    {
```



```
    father_age = x;
    if (father_age<0)
     {
       throw new WrongAge Exception();
     }
  }
}

class Son extends Father
 {
   int son_age;
   Son (int x, int y) throws AgeException, WrongAge Exception
   {
     Super(x);
     son_age = y;
     if (son_age<0)
     {
       throw new WrongAge Exception();
     }
     if (son_age >= father_age)
     {
       throw new Age Exception();
     }
   }
}

class lab7
{
  public static void main (String args[])
  {
```

```
try
{
Scanner s = new Scanner (System.in);
System.out.println ("Enter father's age :");
int x = s.nextInt();
System.out.println ("Enter son's age :");
int y = s.nextInt();
Son S = new son (x,y);
System.out.println ("Father age is " + S.father_age +
" \n Son age is " + S.son_age);
}
Catch (Wrong Exception wa)
{
System.out.println (wa);
}
Catch (AgeExeption a)
{
System.out.println (a);
}
Catch (Exception e)
{
System.out.println(" enter valid numbers");
}
}
```

```
Output:-
Enter Father's age : 50
Enter Son's age : 20
Father age is 50
Son age is 20

Enter Father's age : 50
Enter Son's age : 70
Age of Son can't be grater than father's age

Enter Father's age : 50
Enter Son's age : -20
age can't be negative

Enter Father's age : y
enter valid numbers
```

30/12/2022

Output: