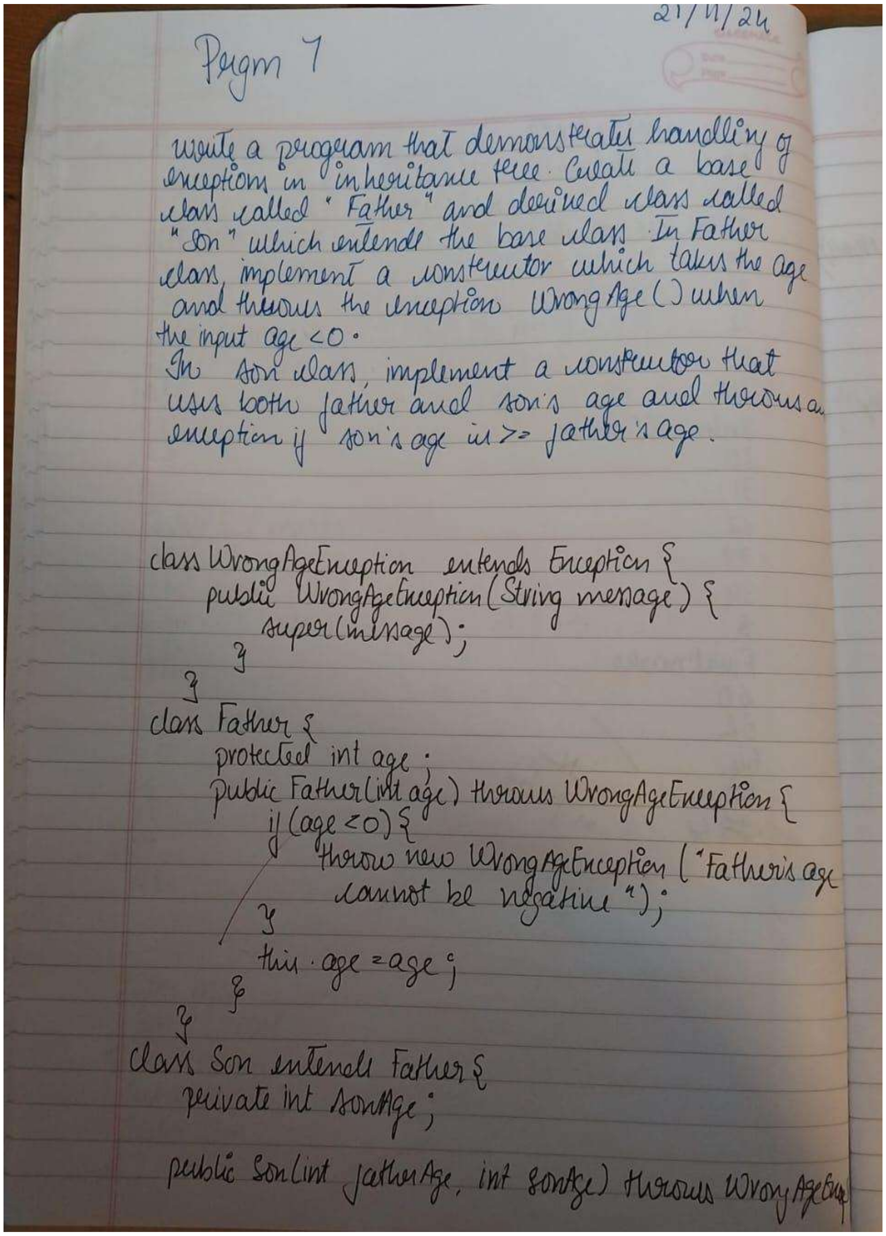
WEEK 7 :

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<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.

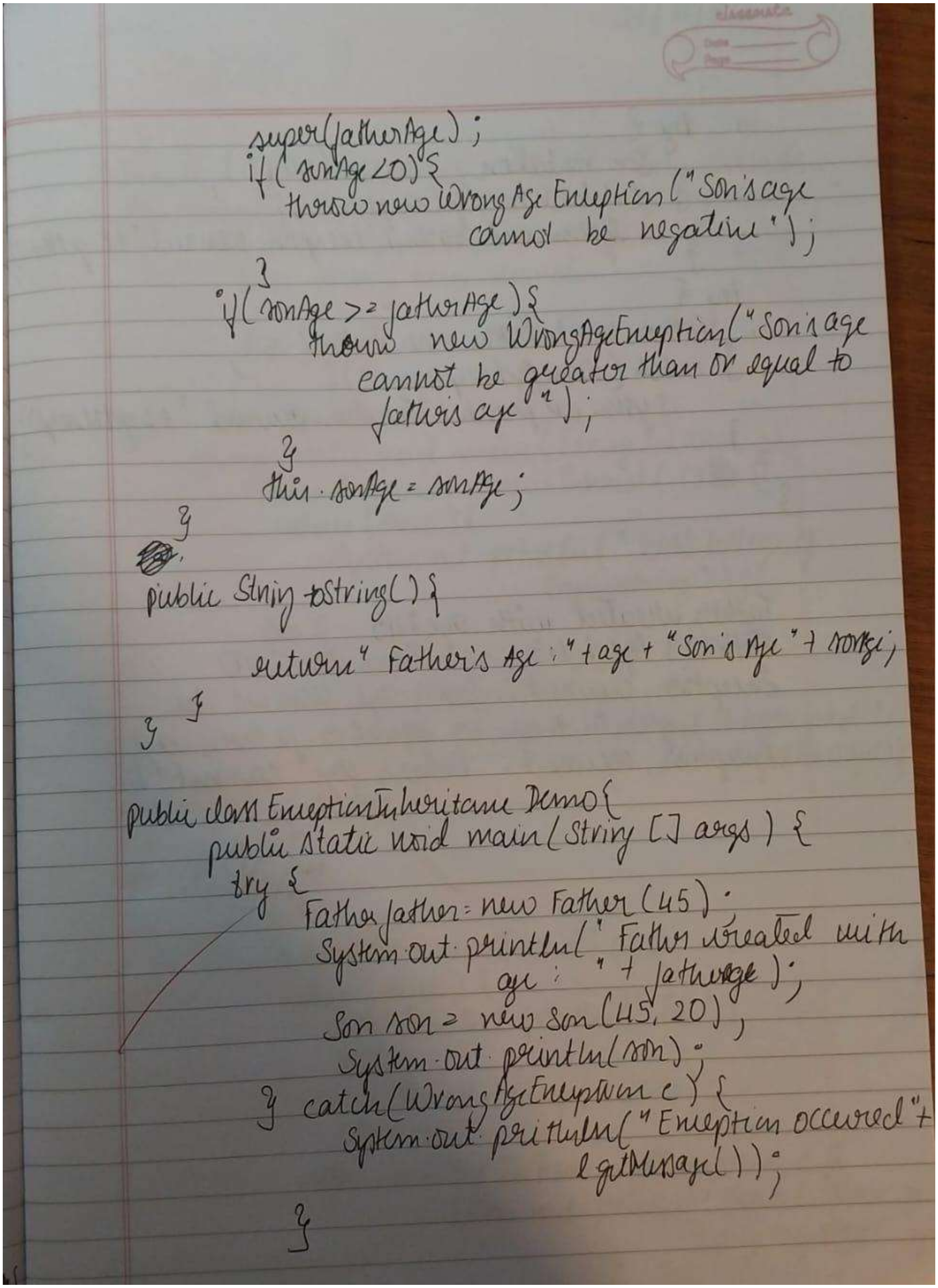


21/11/24

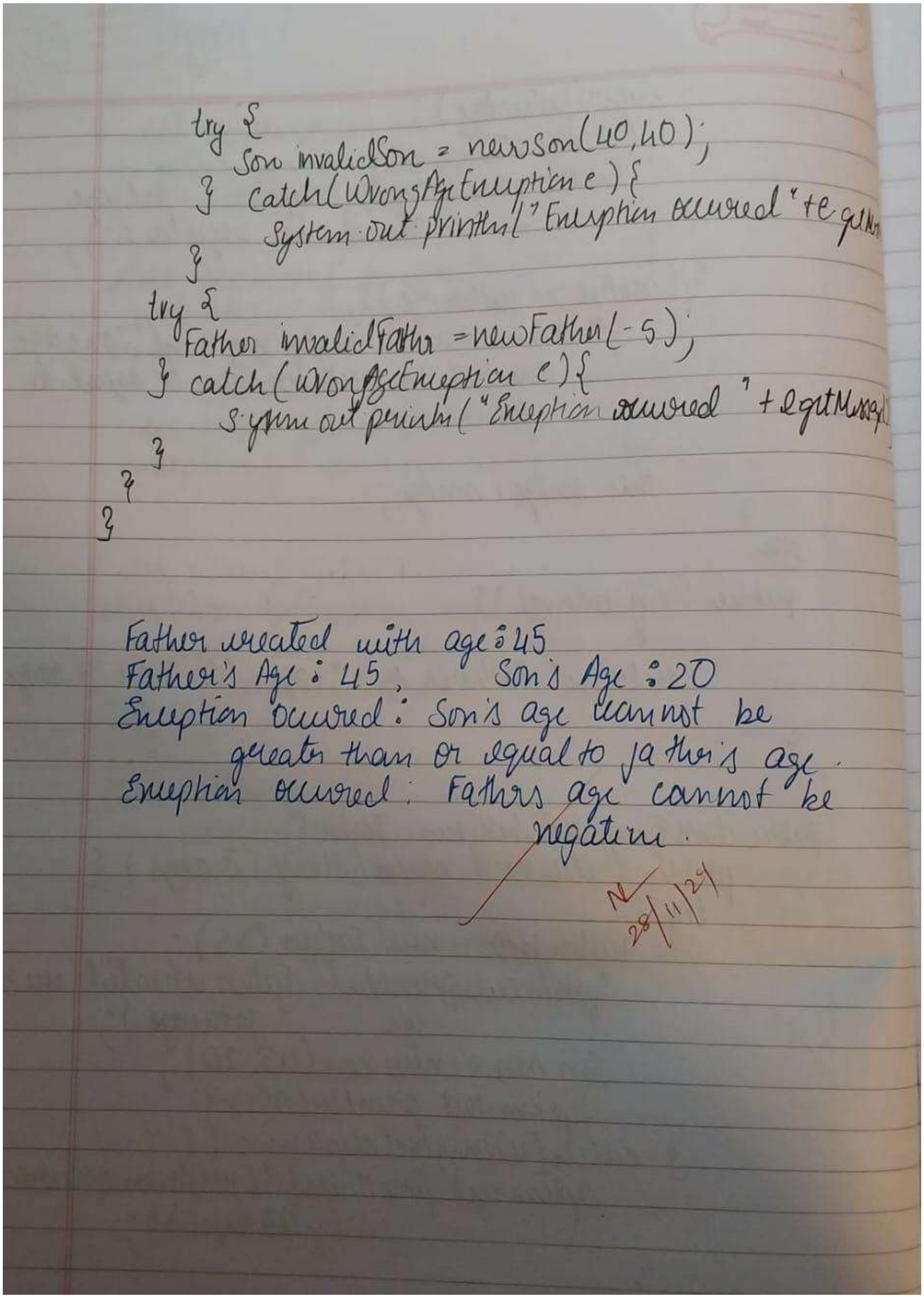
Prgm 7

Write a program that demonstrates handling of exception 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 WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}
class Father {
    protected 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 {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeExc
```



```
super(fatherAge);
if( sonAge<0){
   throw new WrongAgeException("Son's age
                    cannot be negative");
 }
if(sonAge >= fatherAge){
     throw new WrongAgeException("Son's age
          cannot be greater than or equal to
          father's age");
   }
   this.sonAge = sonAge;
 }
public String toString(){
     return" Father's Age :"+age + "Son's Age"+ sonAge;
 }
}

public class ExceptionInheritance Demo{
   public static void main(String [] args) {
     try {
        Father father= new Father(45);
        System.out.println(" Father created with
                 age : " + fatherAge);
        Son son = new Son(45, 20);
         System.out.println(son);
     } catch(WrongAgeException e) {
        System.out.pritnln("Exception occured"+
                        e.getMessage());

     }
```



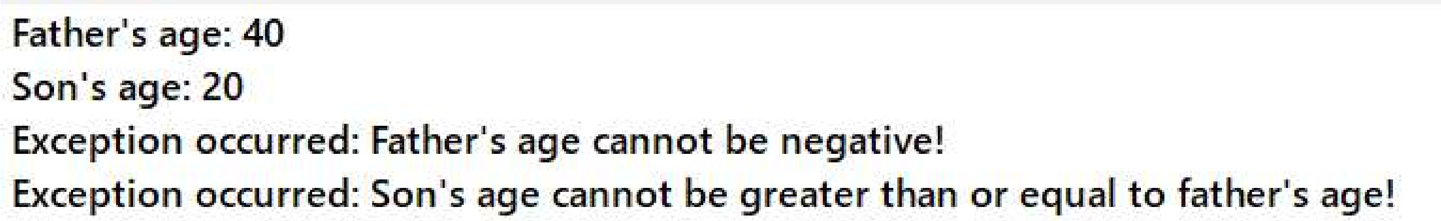
```
        try {
          Son invalidSon = newSon(40,40);
        } catch(WrongAgeException e){
            System.out.println("Exception occured" +e.getMe
        }
      try {
        Father invalidFather =newFather(-5);
      } catch(WrongAgeException e){
          System.out.println("Exception occured" +e.getMessag
      }
    }
}
```

```
Father created with age : 45
Father's Age : 45,    Son's Age : 20
Exception occured: Son's age cannot be
        greater than or equal to father's age.
Exception occured: Fathrs age cannot be
                    negative.
```

28/11/24

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

// Base class Father
class Father {
   int age;

   // Constructor for Father class
   public Father(int age) throws WrongAgeException {
      if (age < 0) {
         throw new WrongAgeException("Father's age cannot be negative!");
      }
      this.age = age;
   }
}

// Derived class Son
class Son extends Father {
   int sonAge;

   // Constructor for Son class
   public Son(int fatherAge, int sonAge) throws WrongAgeException {
      super(fatherAge); // Call the Father constructor
      if (sonAge < 0) {
         throw new WrongAgeException("Son's age cannot be negative!");
      }
      if (sonAge >= fatherAge) {
         throw new WrongAgeException("Son's age cannot be greater than or equal to father's age!");
      }
      this.sonAge = sonAge;
   }
}
public class ExceptionHandlingInheritance {
   public static void main(String[] args) {
      try {
         // Create a Father object
         Father father = new Father(40);

         // Create a Son object
         Son son = new Son(40, 20);
```

```
            System.out.println("Father's age: " + father.age);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Exception occurred: " + e.getMessage());
        }
        try {
            Father invalidFather = new Father(-5);
        } catch (WrongAgeException e) {
            System.out.println("Exception occurred: " + e.getMessage());
        }

        try {
            Son invalidSon = new Son(30, 35);
        } catch (WrongAgeException e) {
            System.out.println("Exception occurred: " + e.getMessage());
        }
    }
}
```



```
Father's age: 40
Son's age: 20
Exception occurred: Father's age cannot be negative!
Exception occurred: Son's age cannot be greater than or equal to father's age!
```