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.

Source Code :

```
import java.util.Scanner;

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

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

class Father {
     int age;
    public Father(int age) throws WrongAgeException {
        if (age <= 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
     int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException,
SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be greater than or
equal to father's age");
        }
        if(sonAge <= 0){
```

```
        throw new WrongAgeException("Wrong age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}


public class FatherSon{
    public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Father's Age: ");
            int fatherAge = sc.nextInt();
            System.out.print("Enter Son's Age: ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Succesfully");
            }
            catch (WrongAgeException e) {
                System.out.println(e.getMessage());
            }
            catch (SonAgeException e) {
                System.out.println(e.getMessage());
            }

       }
  }
```

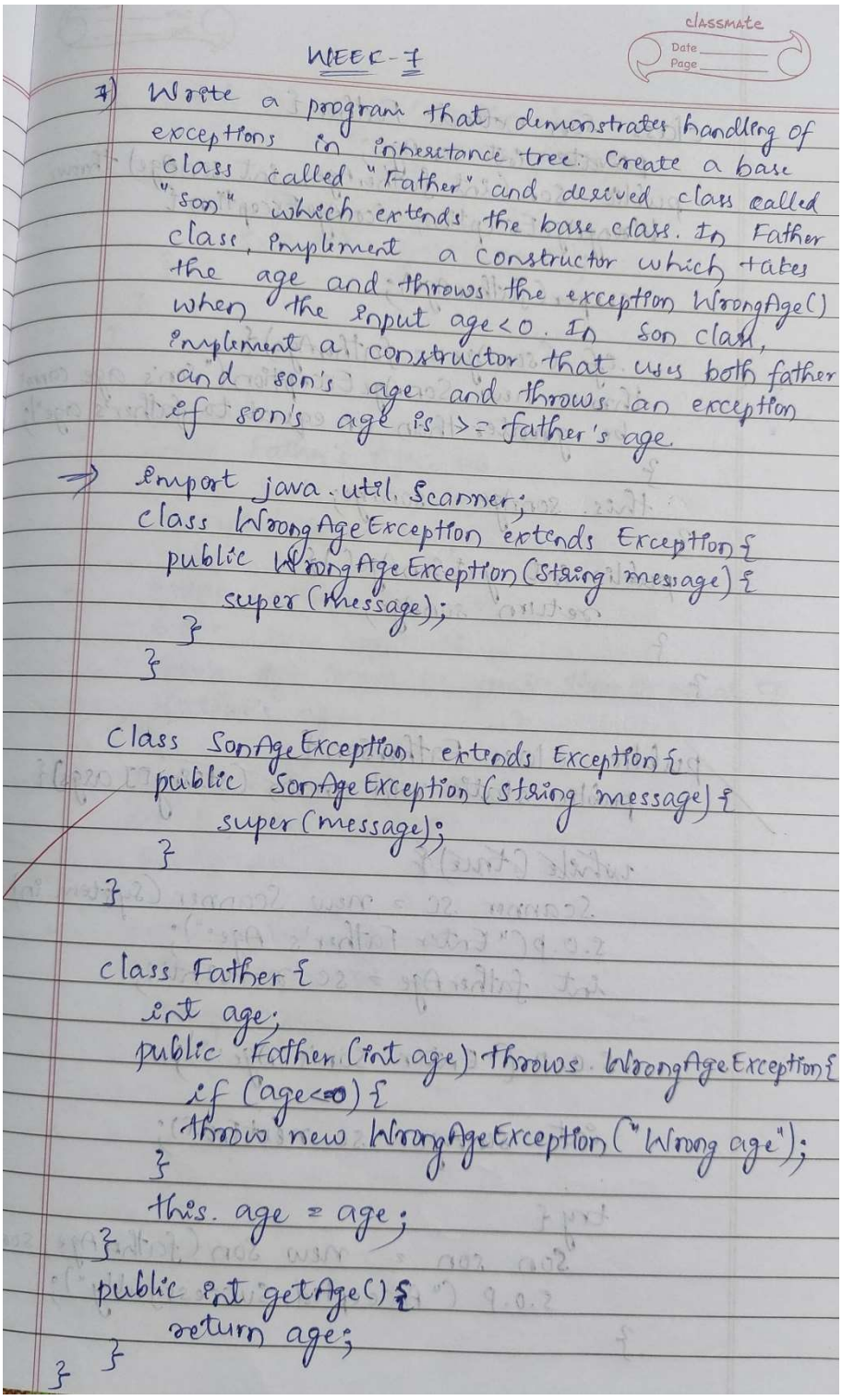
Output :

```
PS C:\Users\satis\OneDrive\Documents\ooj_lab> javac FatherSon.java
PS C:\Users\satis\OneDrive\Documents\ooj_lab> java FatherSon
Enter Father's Age: 44
Enter Son's Age: 26
Accepted Succesfully
PS C:\Users\satis\OneDrive\Documents\ooj_lab> javac FatherSon.java
PS C:\Users\satis\OneDrive\Documents\ooj_lab> java FatherSon
Enter Father's Age: 30
Enter Son's Age: 32
Son's age cannot be greater than or equal to father's age
```

```
PS C:\Users\satis\OneDrive\Documents\ooj_lab> javac FatherSon.java
PS C:\Users\satis\OneDrive\Documents\ooj_lab> java FatherSon
Enter Father's Age: 30
Enter Son's Age: 0
Wrong age
```

```
PS C:\Users\satis\OneDrive\Documents\ooj_lab> javac FatherSon.java
PS C:\Users\satis\OneDrive\Documents\ooj_lab>  java FatherSon
Enter Father's Age: 0
Enter Son's Age: 15
Wrong age
```

Written Code & Output :



WEEK-7

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.

```
import java.util.Scanner;
class WrongAgeException extends Exception{
public WrongAgeException(String message){
    super(message);
  }
}

Class SonAgeException extends Exception{
public SonAgeException(String message){
    super(message);
  }
}

class Father{
  int age;
  public Father(int age) throws WrongAgeException{
    if(age<=0){
    throw new WrongAgeException("Wrong age");
  }
  this.age = age;
}
public int getAge(){
    return age;
  }
}
```

```
class Son extends Father {
    private int sonAge;
    public Son (int fatherAge, int sonAge) throws
    WrongAgeException, SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot
            be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}

public class FatherSon {
    public static void main (String[] args) {
        while (true) {
            Scanner sc = new Scanner (System.in);
            S.O.P("Enter Father's Age:");
            int fatherAge = sc.nextInt();

            S.O.P ("Enter Son's Age:");

            int sonAge = sc.nextInt();

            try {
                Son son = new Son (fatherAge, sonAge);
                S.O.P ("Accepted Successfully");
            }
```

```
catch (WrongAgeException e){
        S.O.P (e.getmessage());
}
catch (SonAgeException e) {
        S.O.P (e.getMessage());
}
}
}
```

Output:

Enter Father's Age: 46
Enter Son's Age: 26
Accepted successfully

Enter Father's Age: 30
Enter Son's Age: 32
Son's age cannot be greater than or equal to father's age

Enter Father's Age: 0
Enter Son's Age: 15
Wrong age

28/11/24

Enter Father's Age: 30
Enter son's Age: 0
Wrong age