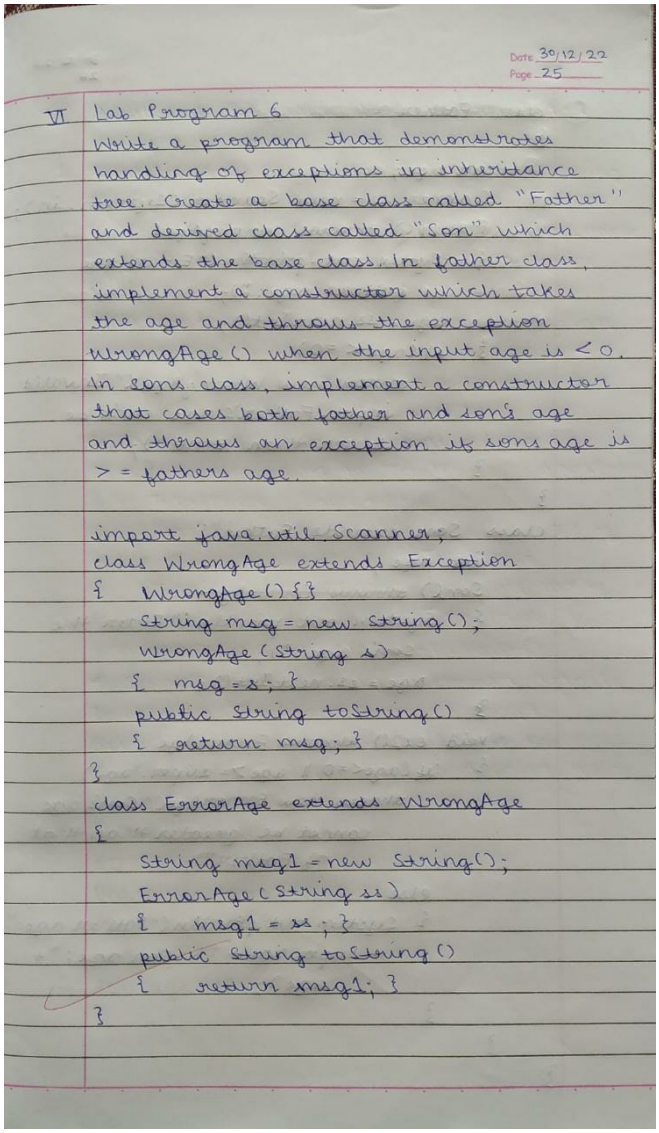
LAB PROGRAM 6:

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and a 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.

CODE:



Date 30/12/22
Page 25

VI Lab 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 is < 0. In sons class, implement a constructor that cases both father and son's age and throws an exception if sons age is >= fathers age.

```
import java.util.Scanner;
class WrongAge extends Exception
{   WrongAge(){}
    String msg = new String();
    WrongAge(String s)
    {  msg = s; }
    public String toString()
    {  return msg; }
}
class ErrorAge extends WrongAge
{
    String msg1 = new String();
    ErrorAge(String ss)
    {   msg1 = ss; }
    public String toString()
    {   return msg1; }
}
```

Date 30/12/22
Page 26

```
class Father
{
    int age;
    Scanner ss = new Scanner (System.in);
    Father() throws WrongAge{
    {  System.out.println("Enter the
          fathers age:");
       age = ss.nextInt();
    }  if(age <=0)
          throw new WrongAge("Invalid
          input. Father's age can't be
          lessen than zero");
    }
}
class Son extends Father {
    int age;
    Son() throws ErrorAge
    {  System.out.println("Enter the
          son's age:");
       age = ss.nextInt();
    }
    void ex2() throws ErrorAge
    {  if(age<=0 || age>= super.age)
          throw new ErrorAge("Son's age
             cannot be greater than that
                of Father");
       else
       { System.out.println("Father age:"
          + super.age + "\n Son age:" +
             age);
       }
    }
}
```

Date 30/12/22
Page 27

```
class A_main {
    public static void main (String xx[])
    {
        Son s = new Son();
        try
        { s.ex1(); }
        catch (WrongAge e)
        { System.out.println(e); }
        try
        { s.ex2(); }
        catch (ErrorAge ea)
        { System.out.println(ea); }
    }
}
```

OUTPUT:

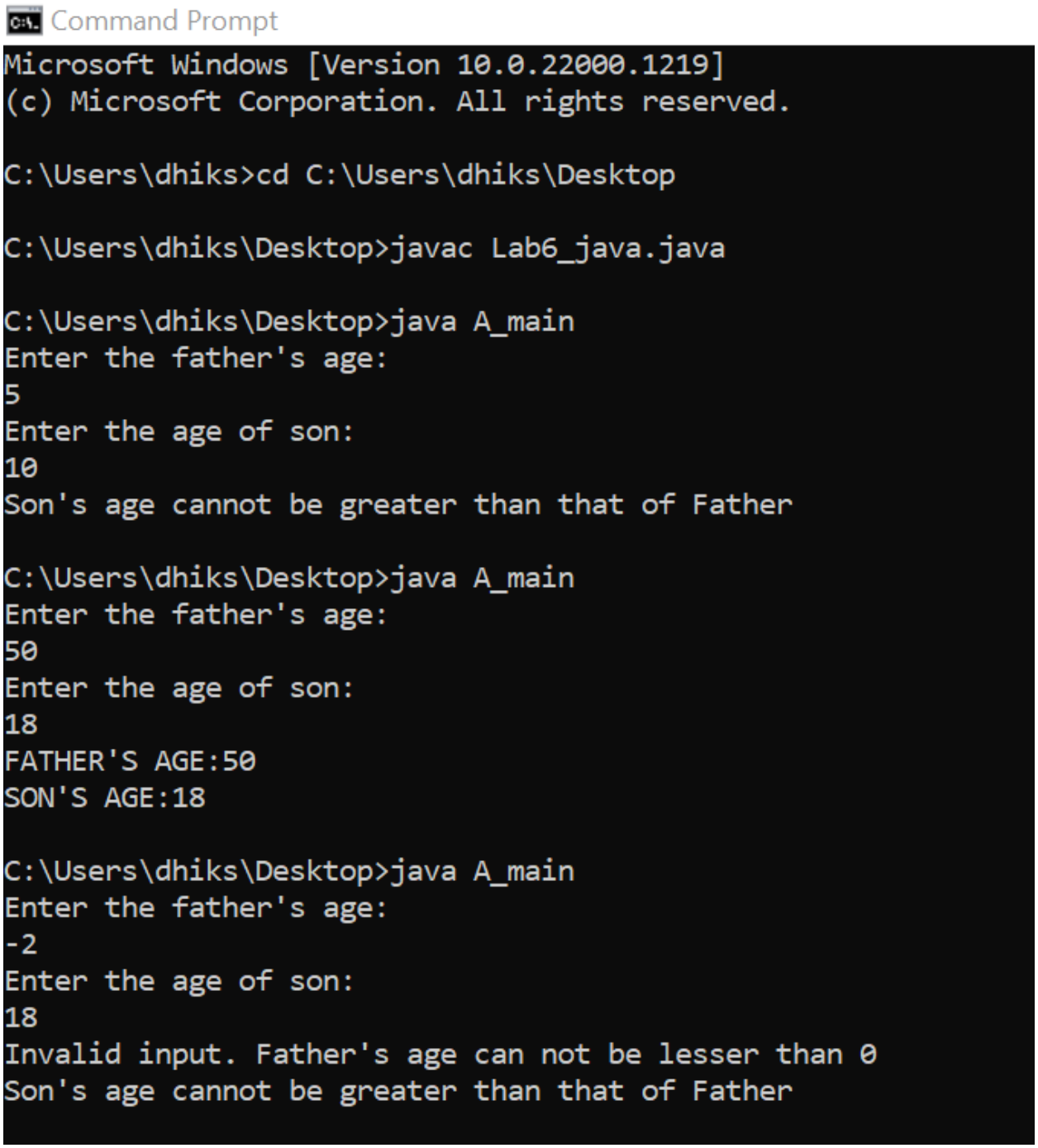
① Enter the father's age:
5
Enter the Son's age:
10
Sons age cannot be greater than that of Father

② Enter the fathers age:
50
Enter the Son's age:
18
Father age: 50
Son age: 18

Date 30/12/22
Page 28

(3) Enter the father's age:
0
Enter the son's age:
18
Invalid input. Father's age cannot be lesser than zero.
Sons age cannot be greater than that of Father

OUTPUT:


Command Prompt
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.
C:\Users\dhiks>cd C:\Users\dhiks\Desktop
C:\Users\dhiks\Desktop>javac Lab6_java.java
C:\Users\dhiks\Desktop>java A_main
Enter the father's age:
5
Enter the age of son:
10
Son's age cannot be greater than that of Father
C:\Users\dhiks\Desktop>java A_main
Enter the father's age:
50
Enter the age of son:
18
FATHER'S AGE:50
SON'S AGE:18
C:\Users\dhiks\Desktop>java A_main
Enter the father's age:
-2
Enter the age of son:
18
Invalid input. Father's age can not be lesser than 0
Son's age cannot be greater than that of Father