Lab — (8) — 30/1/2024   Q. write a program that
                                    demonstrates handling of
                                    exceptions in inheritance tree

Source code :

```java
import java.util.Scanner;
class WrongAge extends Exception
{
    WrongAge (String o)
    {
        super(o);
    }
}

class inputScanner
{
    Scanner s;
    inputScanner()
    {
        s = new Scanner(System.in);
    }
}

class Father extends inputScanner
{
    int fatherAge;
    Father() throws WrongAge
    {
        System.out.println("Enter the age of the
            Father:");
        fatherAge = s.nextInt();
        if (fatherAge < 0)
        {
            throw new WrongAge("Age cannot
                be negative");
        }
    }
}
```

```java
    void display()
    {
        System.out.println("The father's age is" +
        fatherAge));
    }
}
class Son extends Father
{
    int sonAge;
    Son() throws WrongAge
    {
        System.out.println("Enter the age of the
        son");
        sonAge = s.nextInt();
        if(sonAge > fatherAge)
        {
            throw new WrongAge("son's age
            cannot be greater than father's age");
        }
        else if(sonAge < 0)
        {
            throw new WrongAge("Age cannot
            be negative");
        }
    }
    void display()
    {
        System.out.println("The son's age is" +
        sonAge));
    }
}
```

```java
class Main
{
    public static void main(String args[])
    {
        System.out.println("Creating a father class
            object.");
        try
        {
            Father fatherobject = new Father();
            fatherobject.display();
        }
        catch (wrongAge e)
        {
            System.out.println(e);
        }
        System.out.println("Creating a Son
            class object.");
        try
        {
            Son sonobject = new Son();
            sonobject.display();
        }
        catch (wrongAge e)
        {
            System.out.println(e);
        }
    }
}
```

Output:

JAVA Main
Creating a Father class object
Enter the age of the Father:
-45
Wrong Age: Age cannot be negative
Creating a son class object
Enter the age of the Father
-54
Wrong Age: Age cannot be negative

JAVA Main
Creating a Father class object
Enter the age of the Father:
43
The father's age is 43
Creating a Son class object
Enter the age of the Father
43
Enter the age of the son:
50
Wrong Age: Son's age cannot be greater than father's
age:

JAVA Main
Creating a Father class object
Enter the age of the Father
43
The father's age is 43
Creating a Son class object
Enter the age of the Father:
50
Enter the age of the son
32
The son's age is 32: