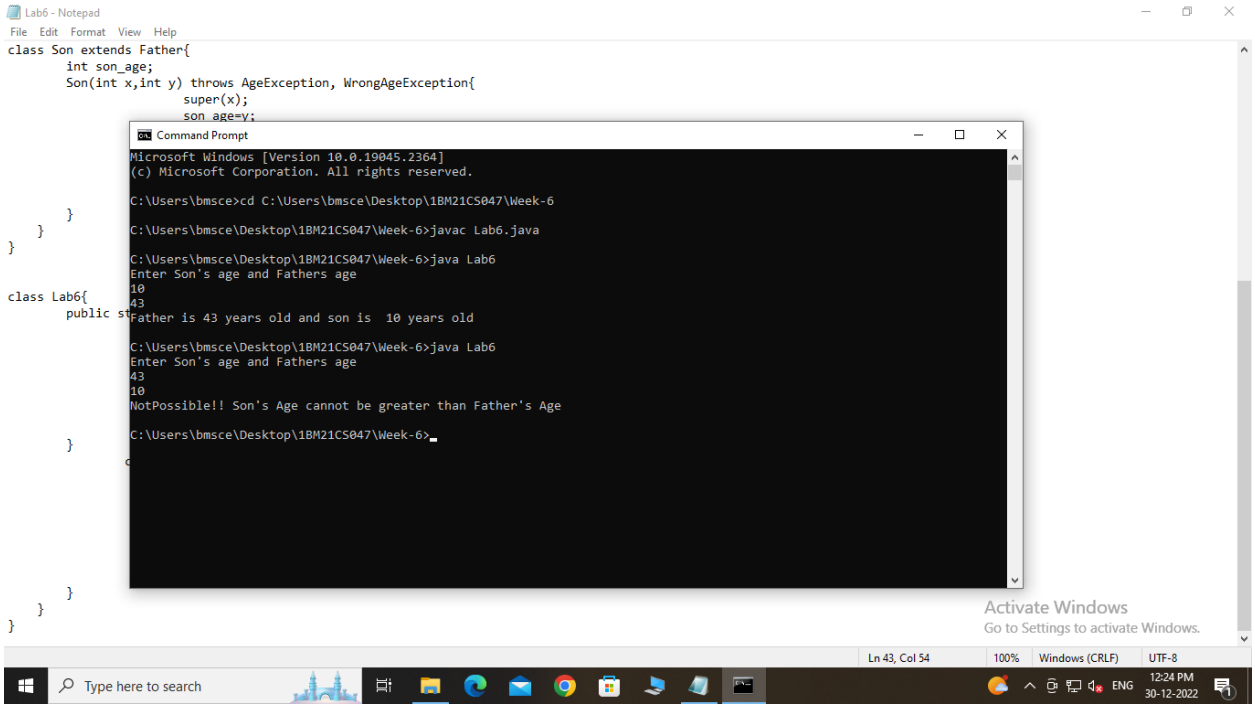
Lab 6:
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. 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.

Program:

```
import java.util.Scanner;
class WrongAgeException extends Exception{
        public String toString(){
                return ("WrongAge!!! Age cannot be negetive");
        }
}

class AgeException extends Exception{
        public String toString(){
                return("NotPossible!! Son's Age cannot 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 Lab6{
	public static void main(String xx[]) {
		try {
			Scanner s=new Scanner(System.in);
			System.out.println("Enter Son's age and Fathers age ");
		int y=s.nextInt();
			int x=s.nextInt();
			Son so=new Son(x,y);
			System.out.println("Father is " + so.father_age + " years old and son is
"+so.son_age + " years old");
	}
		catch (WrongAgeException wa) {
			System.out.println(wa);
		}
		catch (AgeException a){
			System.out.println(a);
		}
		catch (Exception e){
			System.out.println("Age is Interger value");
	}
   }
}
```


Lab6 - Notepad
File Edit Format View Help
class Son extends Father{
int son_age;
Son(int x,int y) throws AgeException, WrongAgeException{
super(x);
son_age=y;
}
}
class Lab6{
public st
}
}
}
Command Prompt
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. All rights reserved.
C:\Users\bmsce>cd C:\Users\bmsce\Desktop\1BM21CS047\Week-6
C:\Users\bmsce\Desktop\1BM21CS047\Week-6>javac Lab6.java
C:\Users\bmsce\Desktop\1BM21CS047\Week-6>java Lab6
Enter Son's age and Fathers age
10
43
Father is 43 years old and son is 10 years old
C:\Users\bmsce\Desktop\1BM21CS047\Week-6>java Lab6
Enter Son's age and Fathers age
43
10
NotPossible!! Son's Age cannot be greater than Father's Age
C:\Users\bmsce\Desktop\1BM21CS047\Week-6>
Activate Windows
Go to Settings to activate Windows.
Ln 43, Col 54
100%
Windows (CRLF)
UTF-8
Type here to search
ENG
12:24 PM
30-12-2022

6. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class "Father" and derived class "son" which extends base class. Implement constructor which takes age in father class, throws exception WrongAge() when input age<0. In son class, constructor which takes both ages & throws exception if son's age is >= father's age.

```
import java.util.Scanner;
Class WrongAge extends Exception{
    public String toString(){
        return("WrongAge !!! Age cannot be negative");
    }
}

class AgeException extends Exception{
    Public String toString(){
        return("Not Possible !! Son's Age cannot be greater
        than Father's Age");

class Father{
    int father_age;
    Father(int x) throws WrongAge{
        father_age = x;
        if (father_age<0){
            throw new WrongAge();
        }
    }
}

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

```
Class Lab6{
    public static void main (String xx[]){
        try{
        Scanner s= new Scanner (System.in);
        System.out.println("Enter Son's age & Father's age");
        int x= s.nextInt();
        int y= s.nextInt();
        Son so = new Son(x,y);
        System.out.println(" Father is " + so.father_age+"
         years old and Son is " + so.son_age + " years old");
        }
        catch (WrongAge wa)
        { System.out.println(wa); }

        Catch (AgeException a)
        { System.out.println(a); }

        Catch(Exception e)
        { System.out.println(" Age is Integer value");}
    }
}
```

Output:

Enter Son's age and Fathers age

10

43

Father is 43 years old and son in 10 years old.

Enter Son's age and Fathers age.

43

10

NotPossible!! Son's Age Cannot be greater than Father's Age

23-12-2022