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.

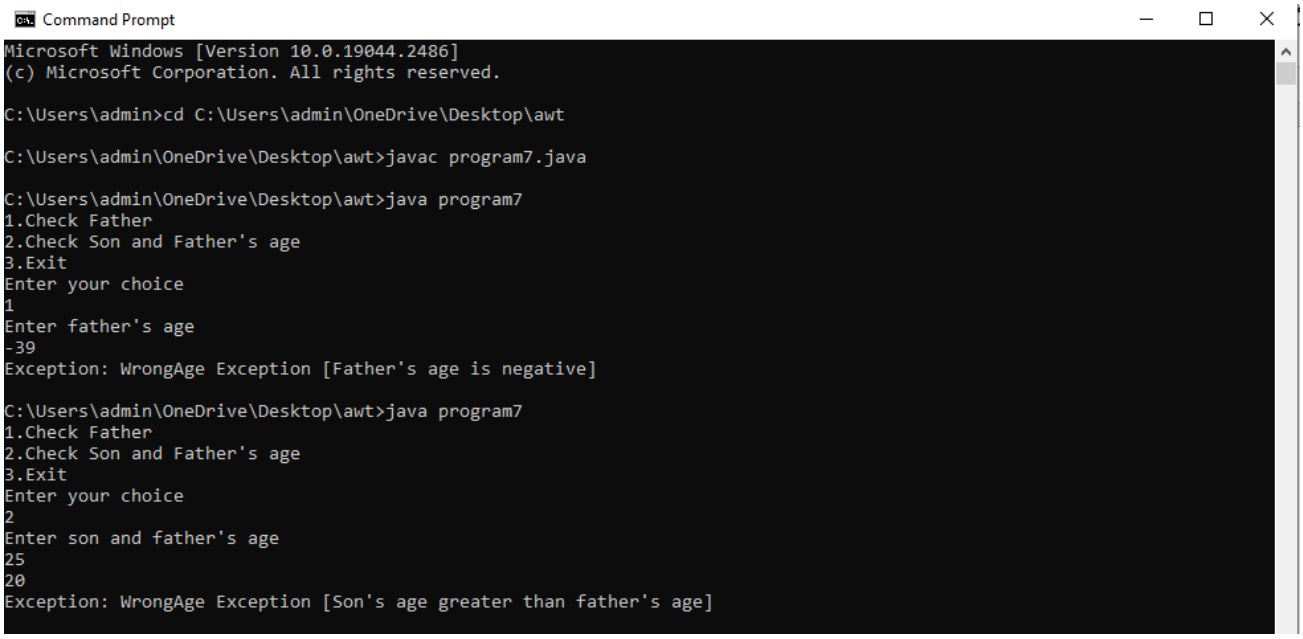
```
import java.lang.*;
import java.util.Scanner;

class WrongAge extends Exception
{
	private String detail;
	WrongAge(String a)
	{
		detail=a;
	}
	public String toString()
	{
		return("WrongAge Exception ["+detail+ "]");
	}
}

class Father {
	int f_age;
	Father(int a) throws Exception
	{
		f_age=a;
		if(f_age<0)
```

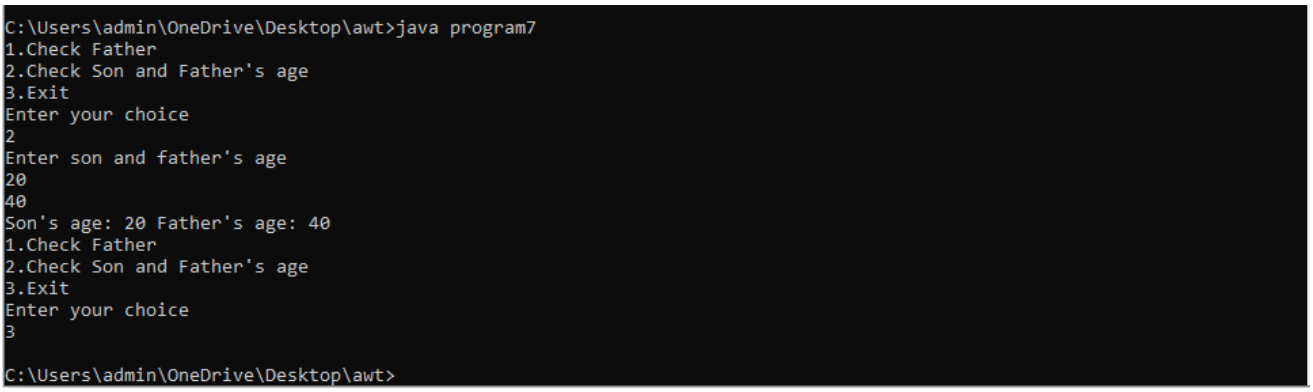
```
            {
                throw new WrongAge("Father's age is negative");
            }

    }
    void display()
    {
        System.out.println("Father's age: "+f_age);
    }
}

class Son extends Father
{
    int s_age;
    Son(int ag,int f) throws Exception
    {
        super(f);
        s_age=ag;
        if(s_age<0)
        {
            throw new WrongAge("Son's age is negative");
        }
        if(s_age>=f_age)
        {
            throw new WrongAge("Son's age greater than father's age");
        }
    }
```

```
	void display()
	{
		System.out.println("Son's age: "+s_age+" Father's age: "+f_age);
	}
}

class program7
{
	public static void main(String args[]) throws Exception
	{
		int ch,f,s;
		Scanner sc= new Scanner(System.in);
		try
		{
		while(true){
			System.out.println("1.Check Father\n2.Check Son and Father's age\n3.Exit\nEnter your choice");
			ch=sc.nextInt();
			switch(ch)
			{
				case 1:
					System.out.println("Enter father's age");
					f=sc.nextInt();
					Father f1=new Father(f);
					f1.display();
					break;
				case 2:
```

```
                                System.out.println("Enter son and father's
age");
                                s=sc.nextInt();
                                f=sc.nextInt();
                                Son s1=new Son(s,f);
                                s1.display();
                                break;
                        case 3:
                                System.exit(0);
                        default:
                                System.out.println("Invalid choice");
                    }
                }
                }
                catch(WrongAge e)
                {
                        System.out.println("Exception: "+e);
                }
        }
}
```

Output:


Command Prompt
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.
C:\Users\admin>cd C:\Users\admin\OneDrive\Desktop\awt
C:\Users\admin\OneDrive\Desktop\awt>javac program7.java
C:\Users\admin\OneDrive\Desktop\awt>java program7
1.Check Father
2.Check Son and Father's age
3.Exit
Enter your choice
1
Enter father's age
-39
Exception: WrongAge Exception [Father's age is negative]
C:\Users\admin\OneDrive\Desktop\awt>java program7
1.Check Father
2.Check Son and Father's age
3.Exit
Enter your choice
2
Enter son and father's age
25
20
Exception: WrongAge Exception [Son's age greater than father's age]


C:\Users\admin\OneDrive\Desktop\awt>java program7
1.Check Father
2.Check Son and Father's age
3.Exit
Enter your choice
2
Enter son and father's age
20
40
Son's age: 20 Father's age: 40
1.Check Father
2.Check Son and Father's age
3.Exit
Enter your choice
3
C:\Users\admin\OneDrive\Desktop\awt>