WEEK 7:

Write a program that demonstrates handling of excep ons 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 excep on WrongAge() when the input age<0. In Son class, implement a constructor that uses both father and son's age and throws an excep on if son's age is >=father's age.

		21/11/24
	Pagm 7	Q au
	woute a program that demon exception in inheritance tree was called "Father" and dere "Son" which entends the base class, implement a constructor and throws the input age <0.	rong Age () when
	The son wan implement a use both father and son anuption if son's age in >=	s age and throusa
	class Wrong AgeEnception entends public Wrong Age Enception (Striv super (missage);	Enception { g memage) {
	clans Father s protected int age: public Fatherlith age) throun if (age < 0) { throw new Wrong/ y comnot be nege	Wrong Age Enception {
	this age = age;	etine ");
u	ais Son entends father & perivate int sonther; public Sonlint Jather Aze, int so	mage) thorous Wrong Agebra
	The second secon	Value III

super(fatherAge);
if (sonAge 20) &
thousa now wrong Age Enuption ("Son's age
commod he negative"); "H(nonAge >= jather Age) &
Thouse new Wrong Agetruption (" Son is age
earn not be queater than or aqual to
jathers are"); This sortige = sontige; public String () } enturn" Father's Age: "+ age + "Son's rye" + rongi; public dom Enception Inheritance Demo (
public Static usid main (String [] angs) {
fry {
Fother latter: New Fother (115). Father Jather: new Father (45).

System out printlul Father wealed with

age: " + Jatherge);

Son son = new son (45, 20), System out printly (son); catch (Wrong Agetruptum c) { Splem out prittyly ("Enception occurred"+ l getheragel));

try &
Son invalication = new Son (40,40);
3 catch (Wrong for Enception e) &
System out printled "Enception excurred" to getter

try &
Father invalid father = new Father (-5);
3 catch (wrong schneption e) {
Symmout printle ("Enception recurred" + logt Mary)

3 Father is Age: 45, Son's Age: 20
Enception occurred: Son's age teamnst be
greater than or equal to ja ther's age
Enception occurred: Fathers age cannot be
negative.

```
class WrongAgeException extends Exception {
  public WrongAgeException(String message) {
     super(message);
  }
}
// Base class Father
class Father {
  int age;
  // Constructor for Father class
  public Father(int age) throws WrongAgeException {
    if (age < 0) {
       throw new WrongAgeException("Father's age cannot be negative!");
    this.age = age;
// Derived class Son
class Son extends Father {
  int sonAge;
  // Constructor for Son class
  public Son(int fatherAge, int sonAge) throws WrongAgeException {
     super(fatherAge); // Call the Father constructor
    if (sonAge < 0) {
       throw new WrongAgeException("Son's age cannot be negative!");
    if (sonAge >= fatherAge) {
       throw new WrongAgeException("Son's age cannot be greater than or equal to father's age!");
    this.sonAge = sonAge;
  }
public class ExceptionHandlingInheritance {
  public static void main(String[] args) {
    try {
       // Create a Father object
       Father father = new Father (40);
       // Create a Son object
       Son son = new Son(40, 20);
```

```
System.out.println("Father's age: " + father.age);
System.out.println("Son's age: " + son.sonAge);
} catch (WrongAgeException e) {
System.out.println("Exception occurred: " + e.getMessage());
} try {
Father invalidFather = new Father(-5);
} catch (WrongAgeException e) {
System.out.println("Exception occurred: " + e.getMessage());
}

try {
Son invalidSon = new Son(30, 35);
} catch (WrongAgeException e) {
System.out.println("Exception occurred: " + e.getMessage());
}

System.out.println("Exception occurred: " + e.getMessage());
}
```

Father's age: 40 Son's age: 20

Exception occurred: Father's age cannot be negative!

Exception occurred: Son's age cannot be greater than or equal to father's age!