```
class WrongAge extends Exception {
    String message;
   WrongAge(String message) {
        this.message = message;
    public String toString() {
        return "WrongAge Exception: " + message;
class Father {
   int fAge;
    Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Father's age cannot be negative!");
        fAge = age;
class Son extends Father {
   int sAge;
    Son(int fAge, int sAge) throws WrongAge {
        super(fAge);
        if (sAge < 0) {
            throw new WrongAge("Son's age cannot be negative!");
        if (sAge >= fAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to
Father's age!");
        this.sAge = sAge;
public class LP7 {
   public static void main(String[] args) {
        try {
            Father father1 = new Father(40);
            Son son1 = new Son(40, 20);
            System.out.println("Father's age: " + father1.fAge + ", Son's age:
" + son1.sAge);
```

```
Father father2 = new Father(-5);
}
catch (WrongAge e) {
    System.out.println(e);
}

try {
    Son son2 = new Son(35, 40);
}
catch (WrongAge e) {
    System.out.println(e);
}

try {
    Son son3 = new Son(50, -10);
}
catch (WrongAge e) {
    System.out.println(e);
}
}
```

```
C:\Users\Sarim Ali\OneDrive\Desktop\OOPs>java LP7
Father's age: 40, Son's age: 20
WrongAge Exception: Father's age cannot be negative!
WrongAge Exception: Son's age cannot be greater than or equal to Father's age!
WrongAge Exception: Son's age cannot be negative!
```

```
Lab-Program-7
 Write a program that demonstrates handling of enceptions,
21/11/24
inher take tree . Chate a base class Called "Father" and des
class "son" which entends the base class. In Father class, implement
a courtly chos which takes the age and theores the enception
wrong Age () when the input age < 0. In Son class implement
a constructor that uses both father I son's age and thous an
enception of son's age is >= fature's age.
clais wrong Age entends Exception &
     Storng msg;
     wrong Age (String msg) &
        this mag = mag;
      public string tostring () {
        return "wrong Age Exception: " + m8g;
class Father &
     int fage;
     Father (Int age) throws wrong Age {
            Hisaw new Wrong Age ("Father's Age can't be nigative")
         rf (age < 0) }
         fage = age;
class son entends Father {
                                                                     Oi
     int sage;
     Son (int fage, Int sage) throws wrong Age &
                                                                    Fai
                                                                    Wo
          super (+ Age);
          if (sAge < 0) {
                                                                   woo
             throw new wrong Age ( som's age cam't be regative');
                                                                   Wood
             throw new wrong og ("Son's age com't be greater them of
          if (SAge >= fAge) {
                                equal to Fathel's Age")
        this · SAge = SAge;
```

public class Toy Catch & public state void main (string E) args) { Bried nent Father ft = new Father (40); Son 81 = new Son (40, 20), System. out. println ("Fatur's Ages" + faifinge , Age: " + 11. shge); Father f2 = new Father (-5); catch (wrong Age e) { System out println (e); toy & Son 82 = new Son (35, 40), Eatch & Wrong Age e) 9 Sys trm out println (e); Son 83 = new Son (50, -10); 2) catch (Wrong Age e) ?

System. out proutince): Outfut: -Father's Age: 40, Sophs Age: 20 Worngage Exception: Father's age can't be regative Wrong Age Exception: Son's age can't be greates than as equal to Father's Age. WrongAge Exception, Son's age cannot be negative