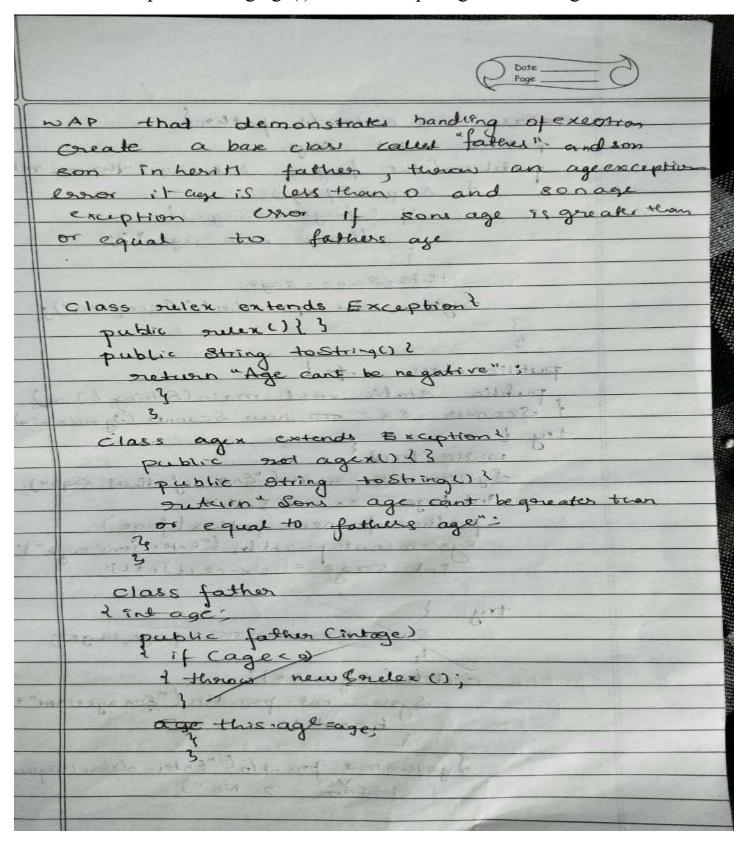
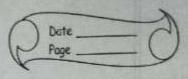
7) Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age=father's age



class son extends jather & public ant sage; publicsofint fage, ent sage) throws relexage, & Super (fage) if Cfage & Sage) two new agents; this. Suge = sage; System out pointly ("Acceptea"); public class 1-15 l public static void main (String [] and Scanner SX = new Scanner (System -in); System out parintial Enter fathers agen) int tage = sx. next Intel; father to new father (fage);

System out print In ("Enter sons age");

Int sage = sx-next Int (); Son 18 = neis son Cfage, sage); catch (agex ex)



2	if (choice!=1)?
	breaking managers
U	tone the second of the second
	3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
	Catch (nelex ex) 1
-	3 2 points
-	finally 2 can't for mail
-	Sx-closer,
-	(A Inf. a points) basedform
1	1 to the first of the state of
-	) += am+
-	
1	of long hav sadist
	Enter fathus age started and
	29
	Enter sons tage to more
	39 Geralgiola Horest
	sons age essor sons age cant regreater
	than or equal to father age
16	general bear to the Tolland from the stante
)	Enter feitige date
	-30
	Age cant be negative

```
import java.util.*;
class relex extends Exception {
  public relex() {}
  public String toString() {
     return "age cant be negative";
  }
}
class agex extends Exception {
  public agex() {}
  public String toString() {
     return "sons age cant be greater than or same as father";
  }
}
class father {
  public int age;
  public father(int age) throws relex {
     if (age < 0) {
        throw new relex();
     this.age = age;
}
class son extends father {
  public son(int fage, int sage) throws agex, relex {
```

```
super(fage);
     if (fage <= sage) {
       throw new agex();
     age = sage;
     System.out.println("accepted details successfully");
  }
}
class fscombine {
  public static void main(String xx[]) {
     Scanner sx = new Scanner(System.in);
     try {
       while (true) {
          System.out.println("enter fathers age");
          int fage = sx.nextInt();
          father f = new father(fage);
          System.out.println("enter sons age");
          int sage = sx.nextInt();
          son s = new son(fage, sage);
          System.out.println("enter details again 1. yes 2. no");
          int ch = sx.nextInt();
          if (ch != 1) {
            break;
          }
     } catch (relex ex) {
        System.out.println("Father's age error: " + ex);
```

```
} catch (agex ex) {
        System.out.println("Son's age error: " + ex);
}
sx.close();
}
```

## Output:

## C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.
C:\Users\hp\Desktop\java>javac fscombine.java
C:\Users\hp\Desktop\java>java fscombine
enter fathers age
23
enter sons age
accepted details successfully
enter details again 1. yes 2. no
enter fathers age
-29
Father's age error: age cant be negative
C:\Users\hp\Desktop\java>java fscombine
enter fathers age
23
enter sons age
45
Son's age error: sons age cant be greater than or same as father
C:\Users\hp\Desktop\java>
```