- 4		
	LAB-PRGM-FIM	
	FATHER AND SON - AGE	
	EXCEPTION 120	
- 15ac	1 we Chara of	1
	Program: Thow exceptions if father age <0 & longe >= father with son's class inheriting Father class':	age
	with son's class inheriting Father class':	
PRAM:	import java retil . Scanner;	
	class Weong Age extends Exception	
	public bleon Age (Sting etc.)	77
	200 (oti): 297 WELL	- 197
	3 super (etc); 297 11211	1
	I have to g	
	0.27 11.01	1)
	class Father 0.08 'S tried !	
	1 Lubrate 3: 12-5	
	int father Age;	
PER IN	Father (int father Age) theory Age	
	B./	
	this father Age = father Age	
	if (father Age < 0)	
	ll	<u> </u>
	theory new bleong Age ("Father's Age sound less than yea"),	
	less than yes	
	9	
-	3	1
	1 2	
e Presidente		

class Son extends Father Son (int father Age, int lon Age) theory allowy Age lupa (father Age); this con Age = con Age; if (ron Age <0) thow new bleary Age (Son's Age connot be less than zuo"); if (lon Age > = father Age) theore new bleng Age ("Son's age cannot be greater than or equal to Father's day");

Class Age Suption Main

E pulli etati void mais (String [] ays)

E Scanne input = now fearner (kanner. in) System out plant ("Inta Father's age:"); sint father Age = input. next Int ();

System-out-print ("Intu lon's Age:"); int lon Age = input. next Int (); Son son = new Son (father Age, eon Age);
System out-pentle ("Tather's and son's age

out valid"); System. out. pernth ("frequention: "+e);

De Enter father's age : 50 htu eon's age: 20 Father's and Son's age are valid 2 Inter father's age: 50 Inter con's age: 60 Sueption: Weary Age: Son's age cannot be greater than or equal to Fatter's age 3 htu futte 's age: -1 htu son's age: 20 Sueption: newy Age: Father's are cannot be less than yea

```
import java.util.Scanner;
class WrongAge extends Exception
{
public WrongAge(String str)
{
super(str);
}
}
class Father
{
int fatherAge;
Father(int fatherAge) throws WrongAge
{
this.fatherAge = fatherAge;
if(fatherAge<0)
throw new WrongAge("Father's Age cannot be less than zero");
}
}
class Son extends Father
{
int sonAge;
Son(int fatherAge,int sonAge) throws WrongAge
{
super(fatherAge);
this.sonAge = sonAge;
```

```
if(sonAge<0)
throw new WrongAge("Son's Age cannot be less than zero");
if(sonAge>=fatherAge)
throw new WrongAge("Son's age cannot be greater than or equal to Father's age");
}
}
class AgeExceptionMain
{
public static void main(String[] args)
{
Scanner input = new Scanner(System.in);
System.out.print("Enter father's age: ");
int fatherAge = input.nextInt();
System.out.print("Enter son's age: ");
int sonAge = input.nextInt();
try{
Son son = new Son(fatherAge,sonAge);
System.out.println("Father's and Son's age are valid");
}catch(WrongAge e){
System.out.println("Exception: " + e);
}
}
}
```

-		1010	
()	ut	nı	IT.
$\mathbf{\circ}$	чı	μ	и.,

Enter father's age: 50

Enter son's age: 20

Father's and Son's age are valid

Enter father's age: 50

Enter son's age: 60

Exception: WrongAge: Son's age cannot be greater than or equal to Father's age

Enter father's age: -1

Enter son's age: 20

Exception: WrongAge: Father's Age cannot be less than zero

Enter father's age: 0

Enter son's age: 20

Exception: WrongAge: Son's age cannot be greater than or equal to Father's age

Enter father's age: 0

Enter son's age: -1

Exception: WrongAge: Son's Age cannot be less than zero