```
C:\Users\bmsce\Desktop\1bm21cs020>java except
Enter the father's age:
90
Enter the age of son:
50

C:\Users\bmsce\Desktop\1bm21cs020>java except
Enter the father's age:
3
Enter the age of son:
6
Son's age is more than father's age

C:\Users\bmsce\Desktop\1bm21cs020>
```

Bafna Gold
Date: Page:

a) 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 takes both father and son's age and throws an exception if son's age is >= father's age.

```
class fatherAgeException extends Exception
{
public String toString() {
  return ("Father's age is less than 0");
}
}
Class sonAge Exception extends Exception {
  int a;
  SonAgeException (int age) {
  a = age;
  }
public String toString () {
if (a<0)
  return ("son's age is lessthan 0");
else
  return ("Son's age is more than father's age");
}
}
class Father {
  int age;
```

```
Scanner in = new Scanner (System.in);
Father(){
System.out.println("Enter the father's age:");
age = in.nextInt();
}
void ex1() throws FatherAgeException
{
  if (age<0)
  throw new FatherAge Exception ();
}
}
class Son extends Father{
int age;
Son(){
System.out.println("enter the age of son:");
age = in.nextInt();
}
void ex2() throws SonAgeException{
   if (age<0 || age > super.age){
   throw new SonAgeException (age);
   }
}
}

public class except{
  public static void main (String[] args){
  Son s = new Son();
   try{
      s.ex1();
   }
  catch (FatherAgeException e){
   System.out.println(e);
  }
```

```
try{
    s.ex2();
}
catch (SonAgeException e){
System.out.println(e);
}
}
}
```

output

Enter father's age:
3
Enter the age of son:
6
Son's age is more than father's age.

Enter the father's age:
90
Enter the age of son:
50