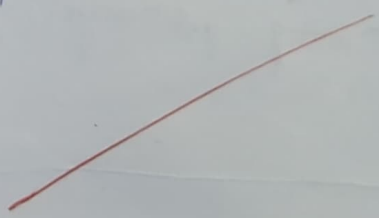
7.

```
import java.util.Scanner;
class Wrongage extends Exception
{
    public Wrongage(String message)
    {
        super(message); this.message = message;
    }
}
class Father
{ int fage;
    public Father(int fage) throws Wrongage
    { if (fage<0)
        { throw new Wrongage("age cant be -ve");
        }
        this.fage=fage;
    }
}
class Son extends Father
{   super(fage);
    int sage; int fage,
    public Son(int sage) throws Wrongage
    { if(fage<sage)
        { throw new Wrongage("Sons age cant be greater than" Fathers age");
        }
        this.sage = sage;
    }
}
```

```
class Fs
{
    public static void main (String args[])
    {
        System.out.println (" Enter the Father & Sons age ");
        Scanner sc = new Scanner (System.in);
        int fa = sc.nextInt();
        int sa = sc.nextInt();
        try
        {
            Son s = new Son( fa, sa )
            System.out.println(" Father age: " + s.fage);
            System.out.println(" Sons age : " + s.sage);
        }
        catch( ~~Arithmetic~~ ArrayIndexOutOfBoundsException e)
        { System.out.println(" Error " + e.getMessage());
        }
    }
}
```



Algorithm

Step 1: Start

Step 2: create class Wrongage which has a ~~void~~ constructor with the string msg; super(msg)

Step 3: Create class Father with int fage.

Step 4: It has a ~~method~~ constructor with int fage which throws class Wrongage

Step 5: if (fage < 0), it throws an exception that says f age cant be negative

Step 6: Create class Son which extends Father. It has a constructor with int sage.

Step 7: if (sage < fage), it throws an exception that says sons age cant be less than fathers age.

Step 8: Create class SF with the main method. Take input of sage & fage. Create objects of class Father & Son. & call the methods inside the try block & print father & Sons age.

Step 9: In the try block print the exception using .getMessage();

Step 10: END

Output: Enter Father & sons age

10 11

Sons age cant be greater than fat