Spring Aop

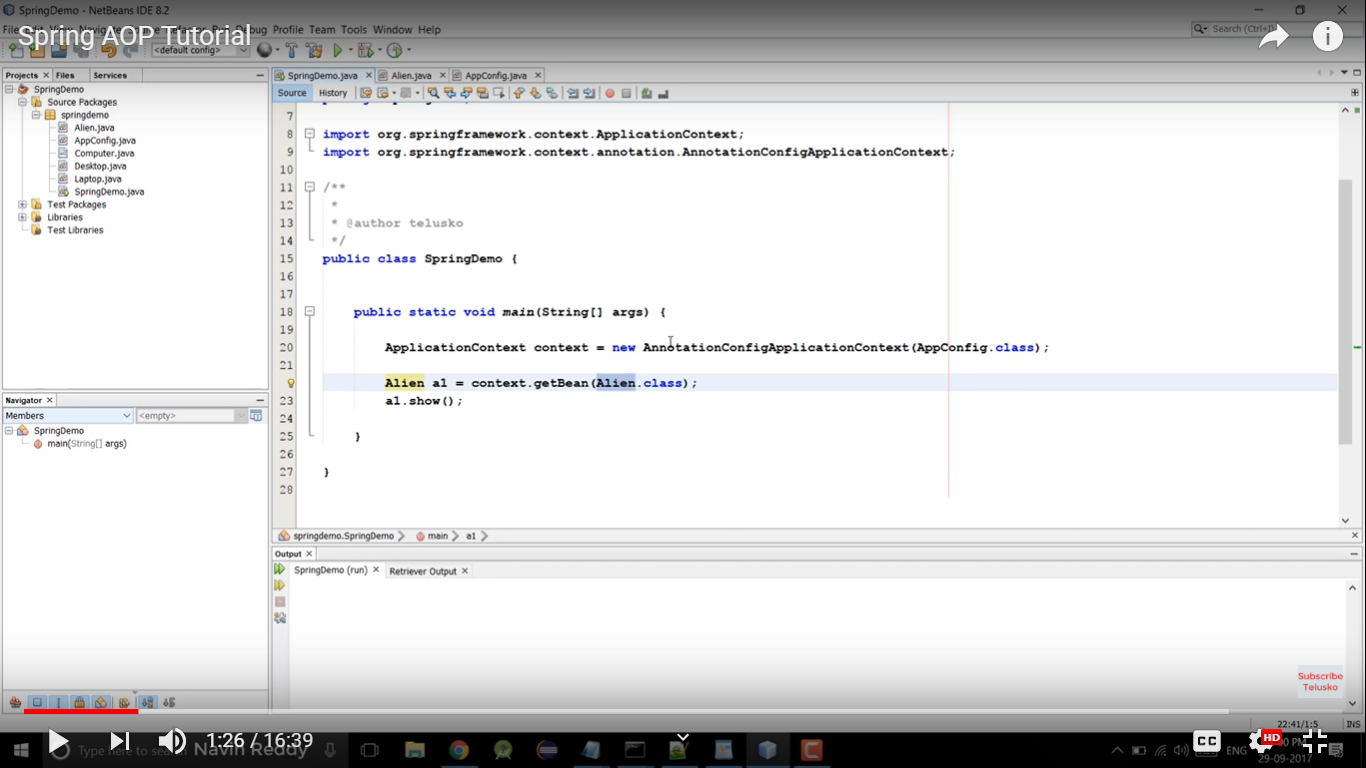
(Aspect Oriented Programing):- Centralized logging krne k kaam aata h (har jagah ni likhte and ek jagah likh k use kr lete h and hum use kr rhe h ye bhi likhte har jagah bs aspect pe likh dete h kon use krne vala h isko)

@Around:-

@EnableAspectJAutoProxy

Telusko Tutorial awesome

ApplicationClasss



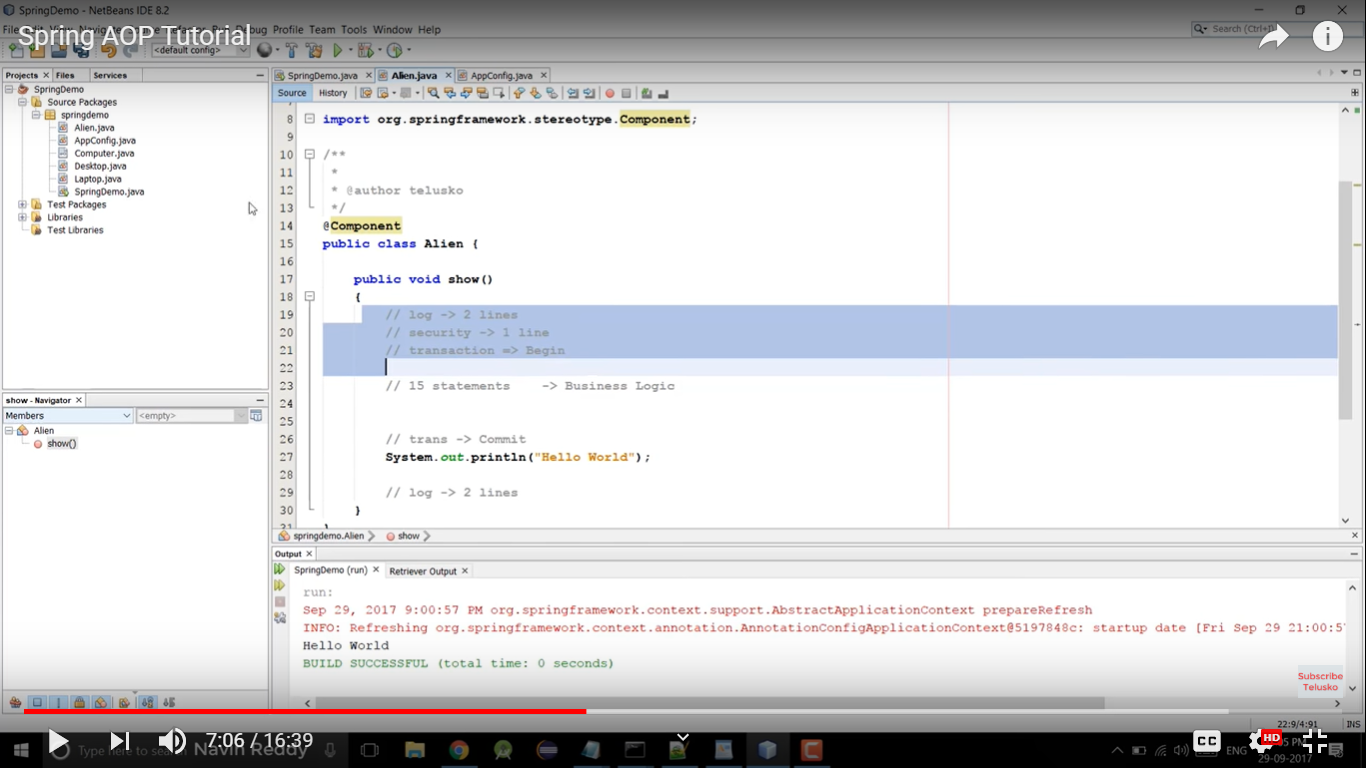
Allien class contains:

//logs ki andar aa gya

//some security check user logged in or not

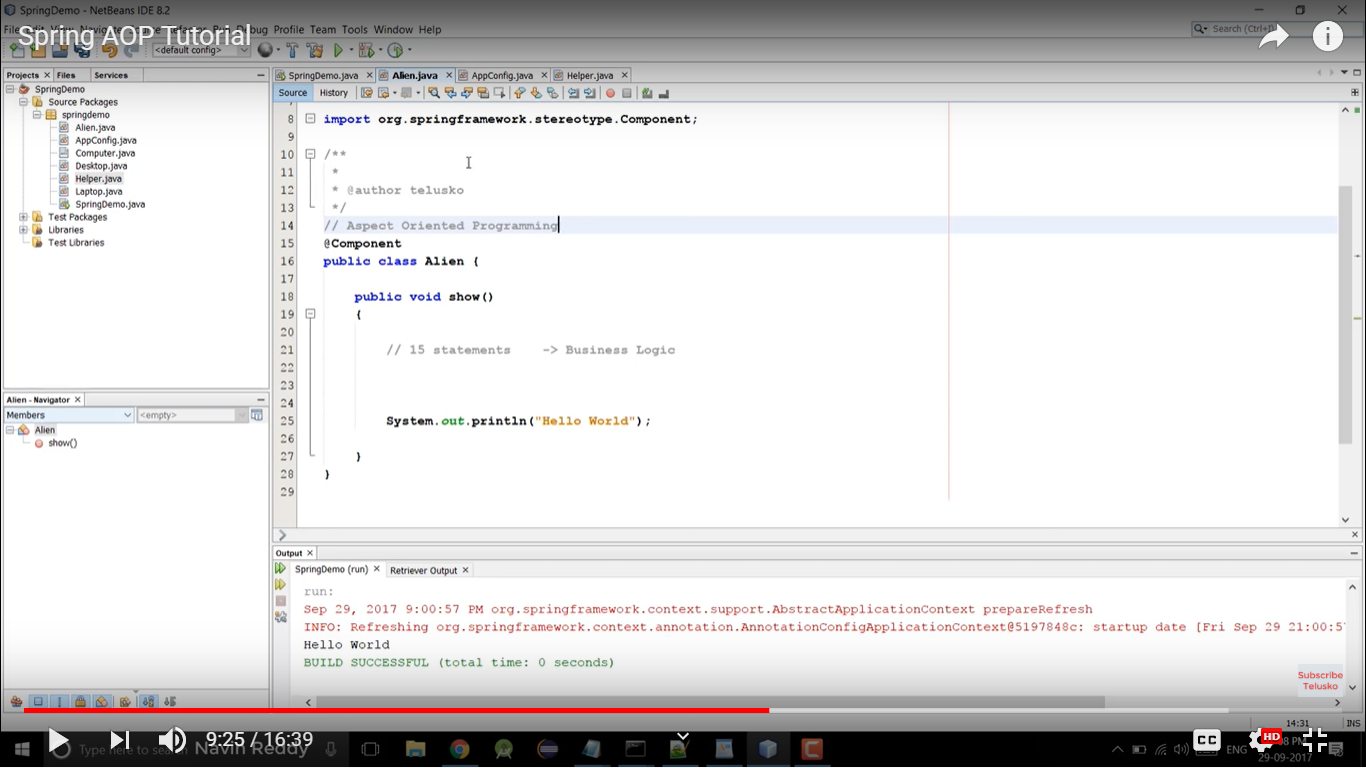
Main Bussiness logic

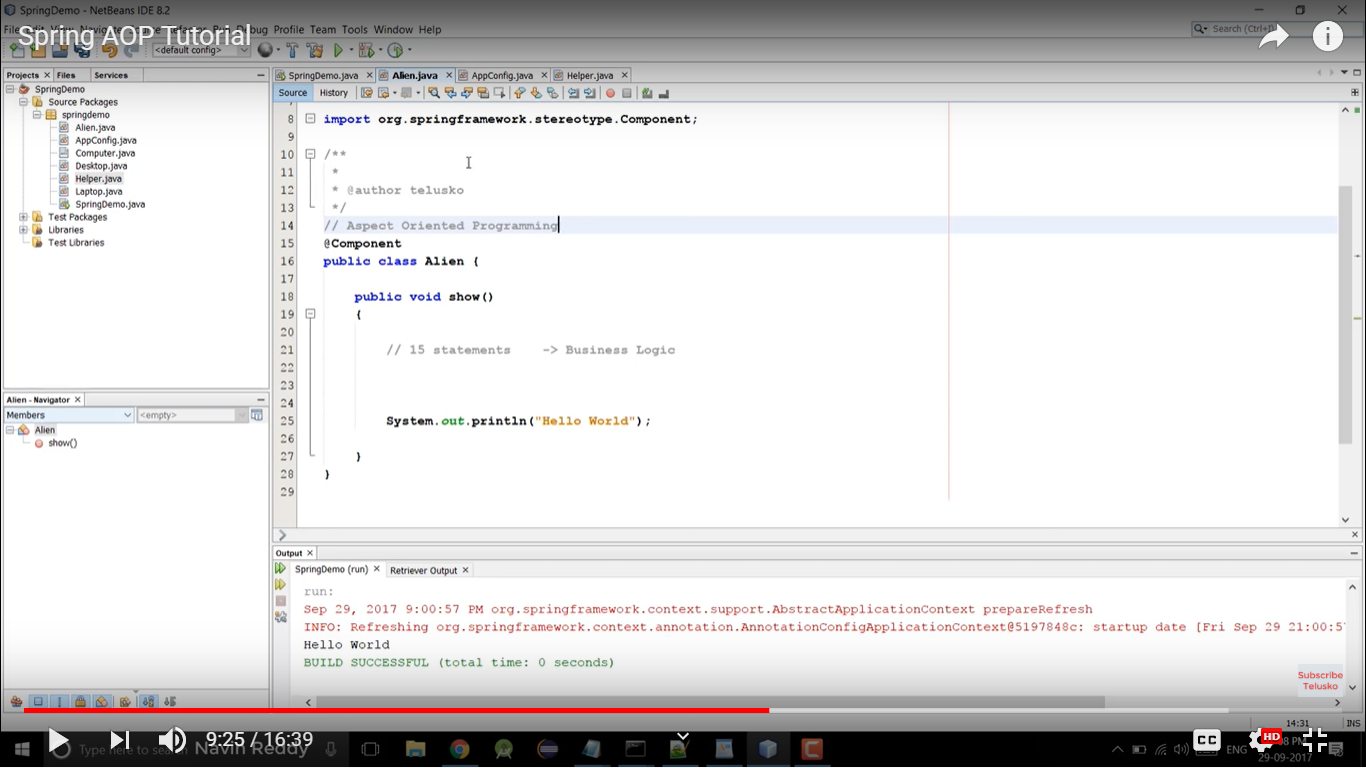
//log ki exit ho gya business se



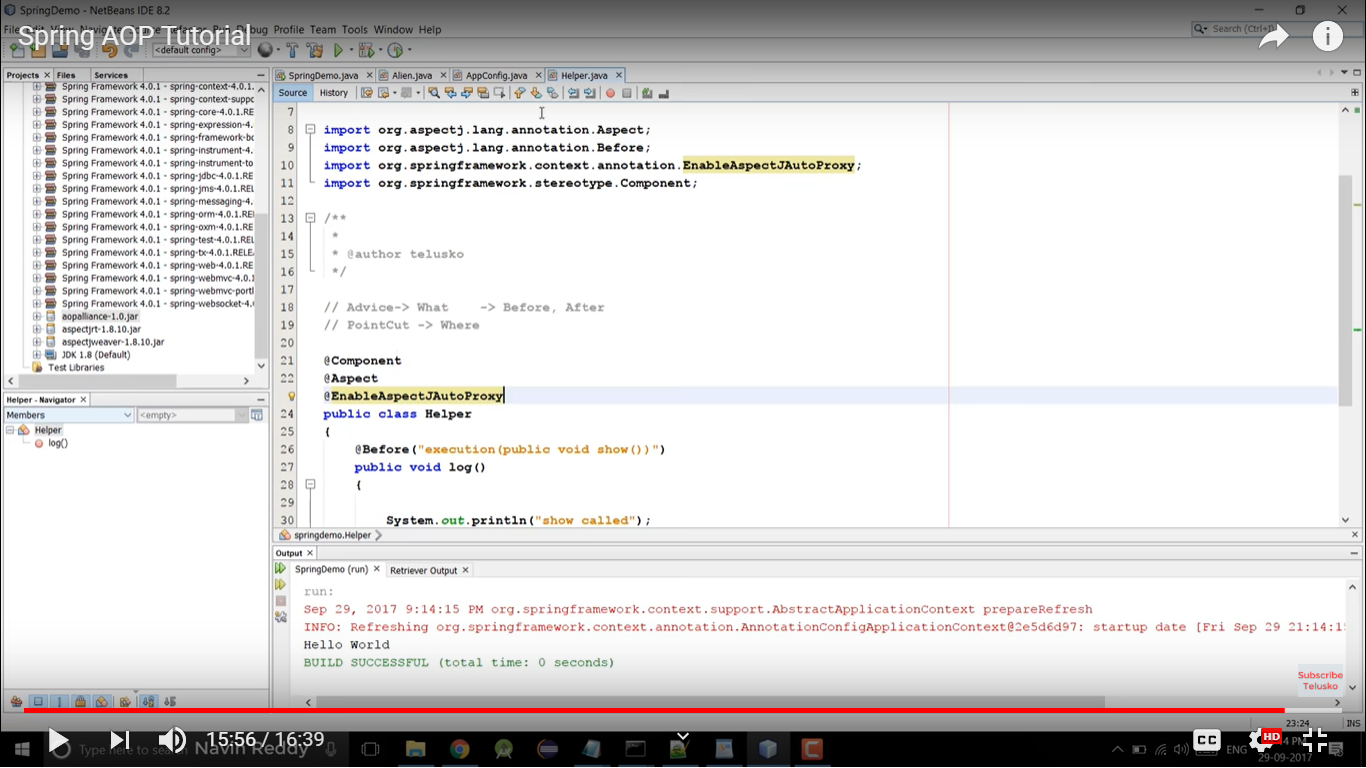
No need to write logs/faltu baat other than business logic every time.. use aop

Aspectj jars needed



Final Alien class

Newly added aspect class



In aspect class we can use 5 advices(when u want this to execute)

1)@Before:- called before entering to method

2.)@After:- called after exiting the method

3.)@Around:- both @Before and @After

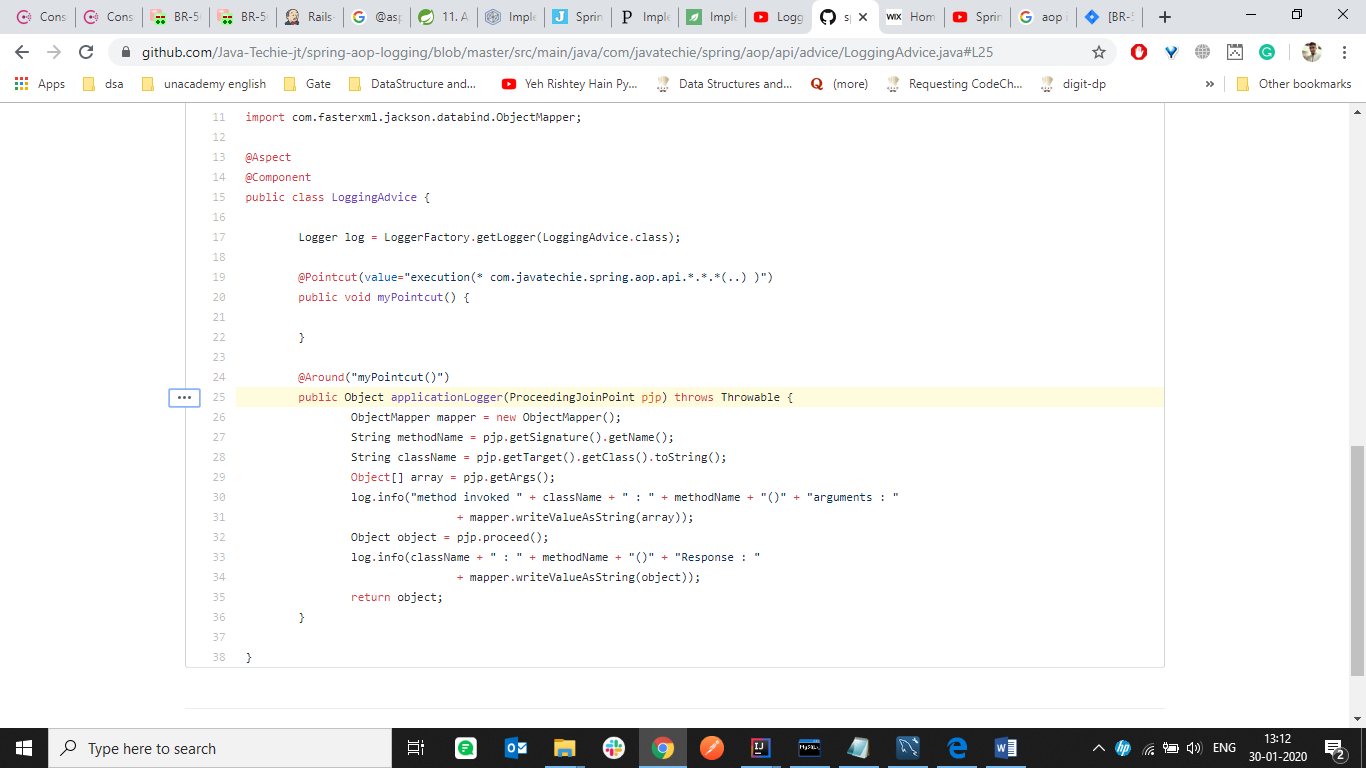
After calling proceedingJointPoint.proceed() hum return value ko capture kr skte h

Object retrun=proceedingJointPoint.proceed(); //return contains retrun value

And Pointcut:- on which functions humko is aspect k method ko call krna h(kin kin methods ka log print krwana h)

@EnableAspectJProxy likhna hota h aspect class k upper or application file m

ProceedingJoinPoint se hum pta kr lenge ki ye log vala logic kis class se call hua, kis method ke liye, or method ne kya parameter as input liye, or method ka retrun type kya h…. etc



@Component is Generic for @Controller, @Service, @Repository.. nothing makes difference but try to use specific for particular task if possible