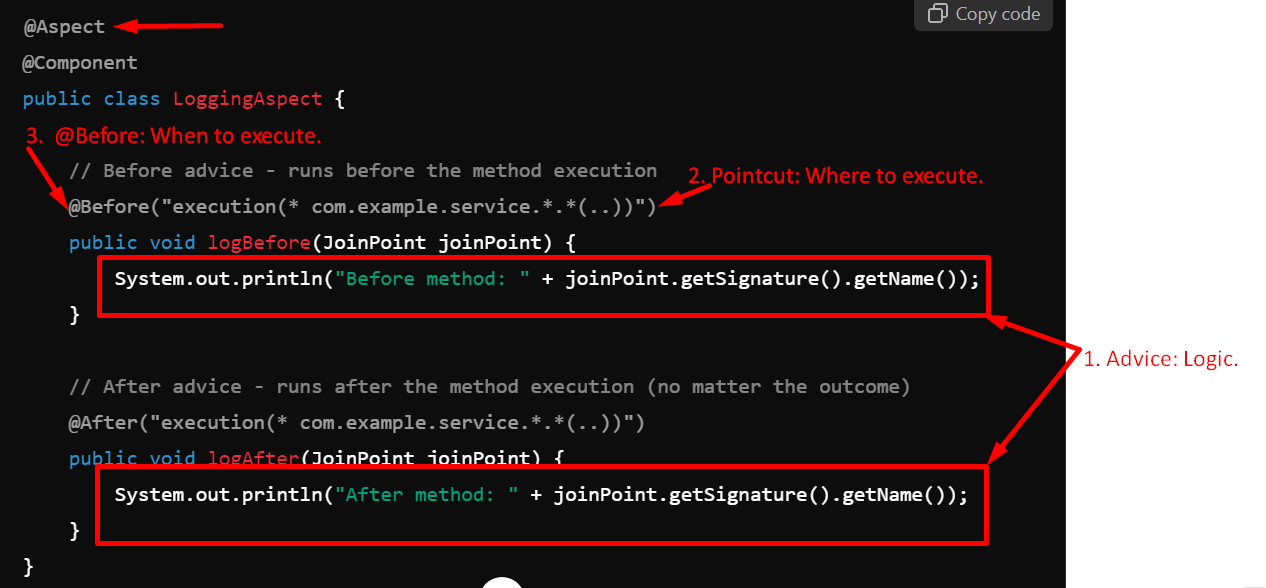
1. **Agenda**:
   1. Jargons around AOP.
2. Jatin: Before moving further into the lecture by the instructor which is a little confusing. Let’s explore the Jargon by ChatGPT.
   1. **Aspect**: A modular unit actually a class having all the advices (logic) and pointcuts(where).  
      
   2. **Advice**: Piece of code to be executed and when to be executed @Before, @After etc.  
      Actually “when” is recognized as types of Advice 😊
   3. Joinpoint (Where): A specific point in app execution where an advice can be applied.  
      There are many kinds of joinPoints such as
      1. Method Calls.
      2. Constructor Calls.
      3. When an exception is thrown.
      4. Object property is set.   
         **NOTE**: In Spring, only Method Joinpoint is supported.
   4. **Pointcut:** Expression which recognises the jointpoints where advice will be applied.

**Where** to execute 🡪 “execute(\* com.example.service.\*.\*(..))”.   
WOW 😊 😊 😊 ChatGPT 😊 😊😊

1. Jargons:  
   Graphical user interface, text, application

   Description automatically generated
   1. **Aspect (What)**:
      1. First you need to identify the aspect meaning which logic you want to execute when a specific method is called.
      2. Logic may be about logging, security or auditing etc.
      3. This is the fundamental part when using AOP.
   2. **Advice (When)**:
      1. After aspect(logic) is decided, you need to decide when to execute such as before method execution or after method execution.
      2. So, we advise Spring Framework when to execute that aspect.
   3. **Pointcut (Which)**:
      1. Which methods are to be intercepted?
      2. Maybe you have thousands of methods in your app and definitely we don’t want to intercept each and every method for executing aspect.
      3. These methods are called pointcuts.
   4. **Join Point**:
      1. Join points define the triggering event that will execute an aspect. In java app, the triggering point is always is when someone is trying to execute a given business logic method inside your app.  
         So, in Java, join point is always a method.
   5. **Target Object**:
      1. In AOP, we decide which methods of your app are to be intercepted and these methods reside inside a bean object.
      2. So, whatever bean that holds the actual method which have business logic inside it is called Target Object.
2. Let’s take a typical scenario whenever you want to implement AOP.
3. Whenever as a developer we want to implement AOP, we can recollect the following statement.  
   Graphical user interface, text, application

   Description automatically generated