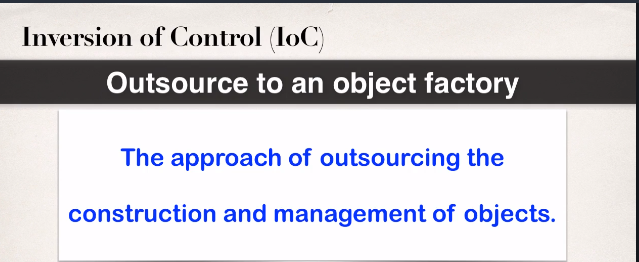
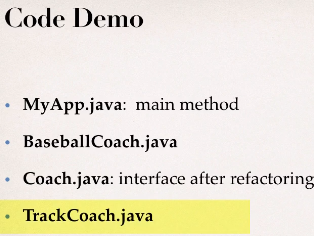
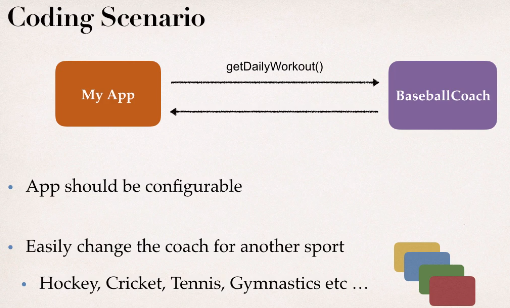
1. 
2. It is simply **design process** of externalizing the management of your objects. That outsource would be handled by an object factory and that is the big idea of “**Inversion of Control**”. 
3. 
4. Let’s go ahead and start with some coding scenario.
5. So, real simple example.We're gonna have our appthat will make use of a coach like a baseball coachand so our app will say hey, baseball coach,give me a daily workout,just so you find out what type of workoutyou need to perform at practice.So, that's the big idea of this application.Now, one thing though is that the management teams saysoh yeah, by the way,the app should be configurable,so this app should easily work for a coachin another sport,so you should easily be able to plug ina hockey coach, cricket coach, tennis, gymnastics, so on.So, they want it to be configurableand also work with any type of coach.So, now you're like oh,I thought the project was easy,now it got a little bit difficult but don't worry,we'll kind of walk through thisand we'll look at some possible solutions for thisand we'll kinda build out a little rough prototype.So, this rough prototype will have four key players.I'll first set off with MyApp.java.It's simply the main program and it has a main method.We'll use BaseballCoach.java.It will be a simple implementation.Then kind of in the version two,once we kind of refactor it,we'll introduce an interface called coach.java,and then we'll also show another implementationcalled TrackCoach.java.Just the whole idea of switching ina different coach and seeing if our applicationcontinues to work.So, those are the main players,that's the big idea.Let's go ahead and move into the next videoand let's get our hands dirtyand let's start writing some code.