1. Agenda:
   1. IoC
   2. DI
2. Graphical user interface, text, application

   Description automatically generated
3. Spring Core is built on principles of inversion of Control which means any project that you pick inside Spring Framework will follow the Software Design Principle which is inversion of control.
4. IoC:
   1. It is a Software Design Principle independent of any Language.
   2. It says never ever create object/dependency in your program directly by yourself instead describe the way of creating an object/dependency by defining configuration and that dependency will be created by external entity.
   3. IoC is just guideline/principle. So what does inversion of control means is instead of programmer controlling the flow of object creation, a framework or external entity should take control of the creation.
5. Dependency Injection (DI):
   1. Pattern through which IoC is achieved.
   2. Spring Framework leverages DI to control the flow of your program with the help of configurations by developer.
   3. Spring core which is base for all Spring projects and Spring Framework is built on core principle call IoC and DI.
6. Brief:
   1. **IoC:** Just a Software Design Principle.
   2. **DI**: It is design pattern which is implementation for IoC.  
      DI shifts the control of creation of objects from application/programmer to an external entity/Spring IoC Container.  
      Objective: To achieve loose coupling b/w components/classes.