**ApplicationContext:**

* Open Demos>Spring Boot>applicationContext in IntelliJ navigate to src>main>java>com.demos>applicationContext
* Create a new POJO for a student with an id and name
* Override the toString method to return "Student{" + "id=" + id + ", name=" + name + "}";
* Create a new class called AppConfig
  + This is how to use the relatively newer “Java Based Configuration” for configuring Java Beans, other options are annotation based (what you’ll use), and xml based (used in legacy projects)
  + Annotate the class with the @Configuration method
  + Create a method called student that returns a Student and annotate it with the @Bean annotation
* In ApplicationContextApplication, create an instance of a new AnnotationConfigApplicationContext and pass in AppConfig.class
* Create a new Student bean via context.getBean(Student.class)
* Sout the student instance and run the program

**Create a Spring Boot Application:**

* Go to [<https://start.spring.io/>](https://start.spring.io/)
* Set “project” to Maven Project
* Set “language” to Java
* Set “Spring Boot” to the most recent supported version
* Set “artifact,” “name,” and the end of “package name” to demoProject
* Set “packaging” to Jar
* Set Java to 17
* Click “add dependencies” and select “Spring Web,” “Spring Data JPA,” and “MySQL Driver”
* Click “generate”
* Unzip the downloaded file into a folder and open it in IntelliJ
* Open the .gitignore file and show students all of the dependencies and packages Spring automatically ignores
* Spring Boot projects also come with maven files
  + mvnw is an executable Unix shell script used in place of a fully installed Maven
  + mvnw.cmd is the Batch version of the above script
  + .mvn is the hidden folder that holds the Maven Wrapper Java library and its properties file
* Create a new POJO called User with a name
* Add the Lombok @Getter and @Setter annotation to the name attribute
* Generate a constructor that sets the name
* Create a class called HelloController and annotate it with @RestController
  + Create a User instance
  + Create a method called greetUser() that returns “Hello “ + user.getName and annotate it with @GetMapping(“/”)
* Run the application with the command ./mvnw spring-boot:run
* Open localhost:8080/greet in the browser