1. 仿写例子（1）（2），叙述基于xml和注解使用ioc的步骤。

基于xml：

**Step1: 添加spring类库。**

**Step2：编写相关类(service及dao接口和实现类)。**

**Step3：编写配置文件beans.xml配置step2中类。**

**Step4：编写测试程序Test。**

注解使用ioc：

**Step1: 添加spring类库。**

**Step2：编写相关类(service及dao接口和实现类)**

**并在实现类相应位置添加注解：**

**dao类前加@Repository**

**service类前加@Service**

**依赖对象前加@Resource**

**Step3：编写配置文件beans.xml。**

**Step4：编写测试程序Test。**

（2）基于注解使用ioc时，controller、service、dao、普通bean使用的注解分别是什么？

注解的value属性用来设置bean在ioc容器中的id，没加时默认值是什么？

**基于注解使用ioc时，controller、service、dao、普通bean使用的注解分别是**

**@Controller、@Service、@Repository、@Component**

**注解的value属性用来设置bean在ioc容器中的id，没加时默认值是类名首字母小写**

（3）依赖的资源前使用的注解有哪两个？它们的区别是什么？

**@Autowired、@Resource**

**区别：@Autowired 默认按类型装配，@Resource默认按名称装配**

（4）从ioc容器中获取bean的语句是什么？

//实例化Spring容器

ApplicationContext ctx = new ClassPathXmlApplicationContext("beans.xml");

//根据id从容器获取Bean对象,默认id为类名首字母变小写

PersonService personService = (PersonService)ctx.getBean("personServiceBean");

（5）bean的作用域（单例、原型等）在xml和注解下如何设置？

**Xml:**

**<bean id="personService" class="service.PersonServiceBean" scope="singleton">**

**<property name="personDao" ref="personDao"/>**

**</bean>**

**不加scope默认就是单例**

**原型 scope=prototype**

**注解：**

通过@Scope注解 默认是单例 @Scope（“singleton”）

原型@Scope(“prototype”)

@Scope(“request”) @Scope(“session”) @Scope(“global”) @Scope(“session”)

（6）看懂IOCDemo实例，理解并叙述ioc基本实现原理。

**IoC**用专门一个容器来创建对象，由Ioc容器来控制对象的创建

例如IOCDemo实例中是用MyIoc来创建的UserService