**[Java基础]**  
1.”==”和equals方法有什么区别？  
答：==是运算符，equals是方法，方法可以通过重写改变其行为，如String的equals就是比较字符串内容。

2. switch语句能否作用在byte上，能否作用在long上，能否作用在String上？  
答： 在Java7之前，switch只能支持 byte、short、char、int或者其对应的封装类以及Enum类型，在Java7中可以支持String。

3. Integer与int的区别？  
答：Integer为包装类，int是基本数据类型。包装类拥有方法和属性，基本数据类型不具备。包装类可以通过intValue来转换成基本数据类型，也可以通过new Integer()将基本数据类型转换为包装类。在JDK1.5后，包装类和基本数据类型可以实现自动转换。

4.try {}里有一个return语句，那么紧跟在这个try后的finally {}里的code会不会被执行，什么时候被执行，在return前还是后?  
答： finally在return之前执行。

5.final, finally, finalize的区别。  
final用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。  
finally是异常处理语句结构的一部分，无论是否异常该部分代码总是执行。  
finalize是Object类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法。

**[Java集合]**  
1.ArrayList、Vector、LinkedList的存储性能和特性？  
ArrayList和Vector都是使用数组方式存储数据，索引数据快而插入/删除数据慢；Vector相对ArrayList是线程安全的，所以性能要低一些；而LinkedList使用双向链表实现存储，插入/删除数据快而索引数据慢。

2.如何对List进行排序？  
使用java.util.Collections的sort静态方法。一种方法是传入一个Comparator对象，另一种方法是List中的对象实现Comparable。

**[多线程]**  
1.synchronized和java.util.concurrent.locks.Lock的异同？  
两者功能类似，都是用来控制线程同步。Lock能完成synchronized所实现的所有功能。Lock需要在finally代码中释放锁，synchronized会自动释放锁。

2.sleep() 和 wait() 有什么区别？  
sleep是线程类（Thread）的方法，传入参数是毫秒数，线程暂停休眠一段时间，到时后会自动恢复，sleep不会释放对象锁。  
wait是Object类的方法，对此对象调用wait方法导致本线程放弃对象锁并等待，只有此对象调用notify或notifyAll后本线程才再次获得对象锁，并运行。

**[JDBC]**  
1.JDBC连接数据库为什么需要Class.forName(“com.mysql.jdbc.Driver”)？  
为了执行Driver中的static代码段，里面注册驱动的代码DriverManager.registerDriver(new Driver())。

2.Statement和PreparedStatement有什么区别？  
PreparedStatement是预编译的SQL，多次执行效率高。  
PreparedStatement可以设置参数，且不用考虑参数中的特殊符号，Statement则要拼SQL字符串，要考虑特殊符号。  
PreparedStatement由于是设置参数而不是拼SQL，可以有效防止SQL注入。

3.什么是数据库事务？JDBC如何实现事务？

数据库事务(Database Transaction) ，是指一系列原子性的操作，要么完整地执行，要么完全地不执行。  
开始时：connection.setAutoCommit(false);  
出现异常时回滚事务：connection.rollback();  
提交事务：connection.commit();

**[HTTP]**  
1.GET和POST有什么区别？  
GET请求参数会在地址栏显示，POST不会。POST提交的数据可以比GET更大，类型更多，例如上传文件需要用POST。POST更安全。  
本质的区别是，GET请求一般没有请求body，参数直接写在URL中，POST请求参数在请求body中。

2.Session和Cookie区别。  
Cookie保存在客户端，而Session保存在服务器上。  
Session一般是通过Cookie中添加一项sessionid来实现功能，但是如果客户端禁用Cookie的话，也可以将sessionid写在url中。  
Session一般关闭浏览器后再打开就无效了，实际上是因为Cookie中的Sessionid失效而不是服务器保存的Session失效。  
Session可以用来做登陆后保持登陆状态，Cookie可以做例如一个月自动登陆这样的功能。

**[Servlet/JSP]**  
1. Servlet生命周期。  
init初始化，整个生命周期只调用一次。  
service处理请求，每次请求调用一次。  
destroy销毁， 整个生命周期只调用一次。

2.JSP内置对象。  
request：请求。  
response：响应。  
out：向客户端写数据的输出流。  
page：该JSP生成的Servlet实例，也就是this，page==this。  
session：客户端和服务器的会话。  
application：存放全局变量，实现用户间数据共享。  
pageContext：可以通过它获取到其他内置对象，它的Attribute只能在被页面访问。  
config：Servlet初始化参数。  
exception：异常。

3.JSP四大属性范围。  
pageContext：作用域是当前页面。  
request：作用域是一次请求。  
session：作用域是一个客户端会话。  
application：作用域是整个应用，所有用户共享。

4.JSP和Servlet的相同和不同。  
相同：  
JSP本质上是Servlet。  
不同：  
用法不同，在MVC模式中，Servlet用来做控制器，用于处理用户请求和业务逻辑，再跳转到相应的JSP，JSP一般用来做页面显示。

5.redirect和forward的区别。  
redirect是服务器发给客户端一个状态码为3XX的响应，由客户端负责跳转，所以浏览器地址栏显示的是跳转后的地址。  
forward又叫转发，是服务器内部的跳转，客户端是不知道的，所以浏览器地址栏显示的是跳转前的地址。

6.JSP两种include有什么区别？  
include指令：<%@include file=”MyJsp.jsp” %>  
可以引用各种文本文件，包括jsp文件，只是单纯的将文件合并，生成Servlet。file是只文件路径，必须是实实在在的文件。  
jsp:include标签：<jsp:include page=”MyJsp.jsp” flush=”true”></jsp:include>  
不是简单的文本合并，而是两个独立的页面。可以理解为将这个页面的运行结果引用进来。page是页面地址，例如可以是Servlet，所以不一定是一个存在的文件，而是一个可以访问的地址。它当然还能带参数，但是include指令不能。

**[Struts]**  
1.Struts 1和Struts 2有什么区别？  
Struts 1的Action只有一个实例来处理请求，需要考虑线程安全问题，Struts 2为每个请求生成一个Action实例，不需要考虑多线程问题；  
Struts 1依赖Servlet API，Struts 2不依赖Servlet API便于单独测试；  
Struts 1采用ActionForm获取参数，Struts 2直接使用Action获取参数；  
Struts 1使用Servlet实现，Struts 2使用过滤器实现。

2.Struts 2 result type。  
dispatcher：默认，forward到jsp页面；  
chain：forward到另一个Action；  
redirect：重定向到jsp页面；  
redirectAction：重定向到另一个Action；  
另外还有其他的类型比如freemarker等。

3.Struts 2如何实现Ajax。  
方法1：在Action的一个方法中直接获取HttpServletResponse，通过输出流直接将结果out.print，缺点是和Servlet API耦合度高，优点是灵活自由。  
方法2：设置result type为stream，通过字符流的方式将字符串设置响应字符串，Struts 2推荐的方法。  
方法3：使用json插件，设置result type为json。