（一）试卷题型

1、判断题（8 小题，8 分）
2、单选题（12 小题，24 分，4 选 1）
3、多选题（4 小题，12 分，5 选 n）
4、问答与分析题（5 / 6 小题，分值不等，24 分）
5、程序填空题（10 个空，20 分）
6、程序设计题（1 题，12 分）

（二）问答与分析题举例

【例 1】为什么“请求转发”会导致通过刷新重复提交表单信息，而重定向不会？请简述理由。
答：（1）请求转发，不会改变浏览器中的地址，即使中间经过了很多其他资源路径，浏览器中仍会保持最初始的访问路径。一旦刷新浏览器，浏览器就会将现在浏览器中的地址，再次发送给服务器。所以，请求转发会导致用户重复提交表单。
（2）重定向，会改变浏览器中的地址。每次重定向，都会将浏览器地址，变为最后访问资源的路径。所以，浏览器即使不断刷新，也只是在不停的访问这个最终资源。因此，重定向不会导致表单的重复提交。

【例 2】为什么关闭浏览器，会话结束？请简述原因。
答：关闭浏览器，浏览器中缓存中的 sessionID 消失。浏览器再次向服务器发送请求时，由于已经没有 sessionID，服务器无法找到对应的 session 对象。session 对象找不到等同于会话结束。

【例 3】为什么 preparedstatement 能够防止 sql 注入攻击？
答：因为 prepareStatement 采用预编译机制。在创建 preparedStatement 对象时即对 SQL 语句进行了预编译，然后传入参数。这时传递过来的参数只被认为是某个字段的值，而不会被识别成一个 sql 指令。例如，' or '1'='1 这样的参数不会被看成是 or 指令，而只是某个字段的值。

【例 4】以下代码用于实现网站访问次数的统计。请指出这段代码的问题（注意：代码中无语法错误），并给出正确的代码。

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
{
    1   HttpSession session = request.getSession();
    2   Integer count = (Integer) session.getAttribute("count");
    3   if (count==null){
    4         count=1;
    5   }else {
    6         count++;
    7   }
    8   session.setAttribute("count",count);
    9   response.getWriter().println("The total number of visits to the website is: "+count);
}
```

答：（1）HttpSession 对象只能实现一次会话内不同请求之间的数据共享，无法实现不同会话之间的数据共享。
（2）应使用 ServletContex 对象存储网站访问次数。

将 1~8 行的代码修改为：

```
ServletContext application = request.getServletContext();
Integer count = (Integer) application.getAttribute("count");
if (count==null){
    count=1;
}else {
    count++;
}
application.setAttribute("count",count);
```

【例 5】某商城，在未登录的情况下，向购物车中放几件商品。关闭浏览器后，再次打开浏览器访问该商城时，购物车中的商品还在，这是怎么做的？

答：将购物车中的商品编号放到 cookie 中，cookie 保存在客户端的硬盘文件中。这样即使关闭浏览器。硬盘上的 cookie 还在。再次打开商城，查看购物车的时候，服务器读取客户端硬盘中存储的 cookie，拿到商品编号，动态展示购物车中的商品。

【例 6】cookie.jsp 文件作用为：在页面中输出浏览器存储的所有 cookie 的名和值。请简述以下代码可能出现的问题，并给出解决方法。

cookie.jsp 代码：

```
<%
    Cookie[] cookies = request.getCookies();
    for (Cookie cookie : cookies) {
        out.println(cookie.getName() + ":" + cookie.getValue() + "<br/>");
    }
%>
```

答：（1）问题：当浏览器中不存储任何 cookie 时，语句 for (Cookie cookie : cookies) {}会报空指针异常。

（2）解决方法：在遍历 cookies 数组之前进行空指针的判断。

```
if (cookies != null) {
    for (Cookie cookie : cookies) {
        …
    }
}
```

【例 7】login.jsp 中，定义了一个复选框，代码如下：

```
<input type="checkbox" name="reb" id="reb" value="y">记住我
```

LoginServlet.java 中，判断是否选中复选框的 Java 代码为：

```
String reb = request.getParameter("reb");
if (reb.equals("y")) {
    System.out.println("用户选择了记住我！") ;
else {
    System.out.println("用户没有选择记住我！");
}
```

请简述 LoginServlet.java 的代码在运行时的问题，以及相应的解决方法。

答：（1）问题：当用户没有选中“记住我”复选框时，reb.equals("y")会报空指针异常。

（2）解决方法：将 reb.equals("y")修改为"y".equals(reb)

【例 8】有如下两段代码：

page1.jsp：

```
<form action="page2.jsp" method="post">
    用户名：<input type="text" name="username"><br>
    密码：<input type="password" name="userpass"><br>
    <input type="checkbox" name="remember" value="yes">记住我<br>
    <input type="submit" value="登录">
</form>
```

page2.jsp：

```
<%
    String username = request.getParameter("username");
    String userpass= request.getParameter("userPass");
    String remember = request.getParameter("remember");
%>
```

若不输入用户名和密码，且未选中“记住我”，单击“登录”按钮，则 username、userpass、remember 的值分别是多少？

答：""、null、null

【分析】不输入用户名和密码，且未选中“记住我”，则传递给 page2.jsp 的参数为：username=&userpass=

因此：

String username = request.getParameter("username"); // 返回 ""

String userpass= request.getParameter("userPass"); //参数名大小写错误，没有与 userPass 对应的数据，返回 null

String remember = request.getParameter("remember"); //没有选中复选框，则参数中没有 remember，因此也返回 null

【例 9】设有如下代码。

page1.jsp：

```
<form action="page2.jsp" method="post">
    用户名：<input type="text" name="username"><br>
    密码：<input type="password" name="userpass"><br>
    性别：<input type="radio" name="gender" value="male">男
        <input type="radio" name="gender" value="female">女<br>
    爱好：<input type="checkbox" name="interests" value="sport" checked>体育
        <input type="checkbox" name="interests" value="music">音乐<br>
    <input type="submit" value="提交">
</form>
```

page2.jsp：

```
<% // 以下代码的 getParameter()中填写的参数名均正确
    out.println("用户名为："+request.getParameter("username")+"<br>");
    out.println("密码为："+request.getParameter("userpass")+"<br>");
    out.println("性别为："+request.getParameter("gender")+"<br>");
    out.println("第一项爱好为："+request.getParameter("interests"));
%>
```

当用户打开 page1.jsp 后，不在页面中进行任何操作，直接点击“提交”按钮，则 page2.jsp 的输出结果是什么？

答：

用户名为：
密码为：
性别为：null
第一项爱好为：sport

【例 10】请给出以下代码的执行结果。

```
<%
   int a=97;
   out.println(a);
   out.write(a);
   response.getWriter().println(a+1);
   response.getWriter().write(a+1);
%>
<%=a+2%>
```

答：98 b 97 a 99

【例 11】在 mysql 数据库中创建 users 表的代码以及 users 表中的数据如下所示：

```
create table users
(user_id int primary key auto_increment,
username varchar(20) unique,
userpass varchar(20),
salary decimal(18,2)
);
```

```
mysql> select * from users;
+---------+----------+----------+----------+
| user_id | username | userpass | salary   |
+---------+----------+----------+----------+
|       1 | lisa     | 111      |  8000.00 |
|       2 | mary     | 222      |  9000.00 |
|       3 | susan    | 333      | 10000.00 |
+---------+----------+----------+----------+
3 rows in set (0.00 sec)
```

现通过以下 java 代码，获取前端传送的用户名和密码；验证成功后，将显示该用户的薪水。请说明代码中出现的 SQL 注入攻击，并给出解决注入攻击的方法。

```
1  Class.forName("com.mysql.jdbc.Driver");
2  String url ="jdbc:mysql://localhost:3306/my_db";
3  String user = "root";   String password = "123456";
4  Connection connection = DriverManager.getConnection(url, user, password);
     //接收前端数据：用户名 username 和密码 userpass
5  String username = req.getParameter("username");
6  String userpass = req.getParameter("userpass");
7  String sql = " select salary from users where username='" + username + "'"
8                                                    +" and userpass='"+userpass+"'" ;
9  Statement statement = connection.createStatement();
10   ResultSet resultSet = statement.executeQuery(sql);
11   resp.setContentType("text/html;charset=utf-8");
12   if (resultSet.next()) {
13         BigDecimal salary = resultSet.getBigDecimal("salary");
                    //salary 在 mysql 为 decimal 类型，在 java 中对应的数据类型为 BigDecimal
14         resp.getWriter().println("您的薪水为："+salary);
15    }else{
16         resp.getWriter().println("用户名或密码不正确！无权查看薪水");
17   }
```

答：

（1）SQL 注入举例：通过前端输入用户名 username：lisa '#，密码 userpass 随意填入，就能查看用户 lisa 的薪水。

（以下为解释，答题时无需填写：

是 mysql 的注释符号，因此在 username 为：lisa '# 时，#后面的代码全部无效，那么被执行的 sql 语句为：

select salary from users where username=' lisa'

即实现根据用户名查找记录的操作。同理，要查看 susan 的薪水，只需前端输入用户名：susan'# 即可。通过上述形式的用户名，成功绕过了密码校验，因此这种用户名也被称作“万能密码”。

感兴趣的同学，可以自行构建前端界面，并在上面提供的后端代码的基础上加上异常处理等代码，从而验证本例的“万能密码”）

（2）解决方法：使用 PreparedStatement 代替 Statement。

将 7-10 行的代码替换为：

```
String sql = "select username,salary from users where username=? and userpass=?";
PreparedStatement preparedStatement = connection.prepareStatement(sql);
preparedStatement.setString(1,username);
preparedStatement.setString(2,userpass);
ResultSet resultSet = preparedStatement.executeQuery();
```

（三）程序题举例

1、 通过反射读取一个类中的所有方法。

2、 HashMap、ArrayList 的基本用法。

3、在一个 JSP 或者 Servlet 中，实现网站访问次数的统计。

注意：

（1）在 JSP 中，直接通过 application 访问 ServletContext 对象。

（2）在 HttpServlet 的子类的 doGet() / doPost()中，获取 ServletContext 对象的代码为：

```
public class CountServlet extends HttpServlet{
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
            ServletContext application = getServletContext();
            //或者：ServletContext application = req.getServletContext();
    }
}
```

3、使用 cookie 和 session，实现七天免登录。详见“实验 9 补充练习”。

4、敏感词过滤。详见“实验 9 补充练习”。

5、记录用户上一次的登录时间。

6、记录用户浏览过的商品信息。

7、使用数据库访问通用类 Dbutils，实现登录和注册功能。

8、使用数据库访问通用类 Dbutils，实现数据的伪删除。

（1）什么是伪删除？

在数据库中，通过 delete 语句删除的记录，无法“一键找回”。为了让网站用户能够快速“找回”误删除的数据，则需要实现数据的伪删除。

伪删除指的是，数据库中不是真正删除数据，而是通过标记将数据标记为已删除。这在需要保留数据历史记录的情况下非常有用。伪删除的实现思路：

- ✧ 在表中增加字段： is_deleted (boolean 型)，并设置 is_deleted 的默认值为 0（逻辑假）
- ✧ 一旦删除记录，则设置该记录的 is_deleted 的值为 1（逻辑真）

  update student set is_deleted=1 where sno='99001' and is_deleted=0 # 删除 99001 的学生记录
- ✧ 选择记录的时候，加上限制条件 is_deleted =0

  select * from student where is_deleted = 0 #浏览 student 表中的全部记录

（2）Mysql 数据库代码（供同学自行操作时使用，读懂即可）

```
create table student
(sno char(5) primary key,
sname varchar(50),
sage int,
is_deleted boolean default 0);
insert into student(sno,sname,sage) values('99001','lisa',20),('99002','mary',21);
```

（3）Servlet 的核心代码

```
Dbutils dbutils = new Dbutils();
String sql = "update student set is_deleted=1 where sno=? and is_deleted=0";
String sno = req.getParameter("sno");
Object [] params = {sno};//将前端输入的学号作为 sql 参数
int i =dbutils.executeUpdate(sql,params);
resp.setContentType("text/html;charset=utf-8");
if (1==i) {
    resp.getWriter().println("成功删除学号为："+sno+" 学生的记录");
}else {
    resp.getWriter().println("该生不存在，删除失败");
}
```