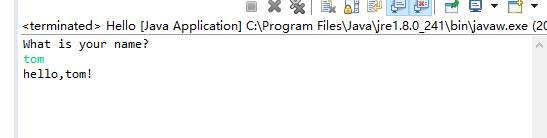
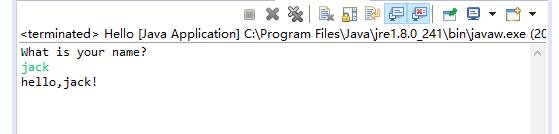
**题目①**

**一、源程序调试过程**  
1、System.out.println()语句报错：忘记以分号结束语句。

2、Scanner in = new Scanner(System.in);语句无法执行：未导入import java.util.Scanner;  
  
**二、实验实习结果分析**  
1、输入“tom”,应输出“hello,tom!”程序运行正确  
2、输入“jack”,应输出“hello,jack!”程序运行正确

3、验证了程序的正确性，确保程序能够在各种环境下运行。  
**三、心得体会**  
通过编写简单的HelloWorld程序，我学会了基本的创建java项目，新建class以及import的方法，对大致的java语言编写程序有了基本的了解和掌握。并且能够运行java项目并成功地在在线系统提交。我巩固了基本的语法知识，并且学会了如何使用Java中的输出语句。这个简单的实验为我后续学习Java编程打下了基础。

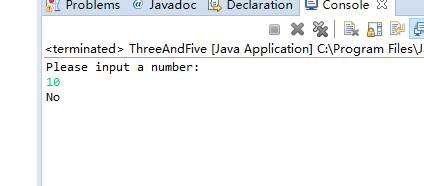
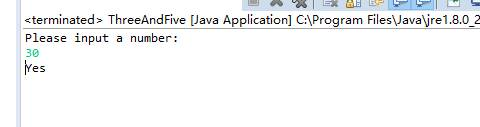
**题目②**

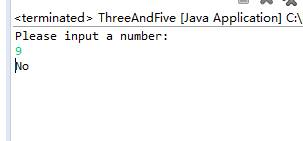
**一、源程序调试过程**  
1、判断整除的模运算报错：“=”是赋值运算符，而“==”才是逻辑运算符，其功能是判断左右基本类型是否相等。

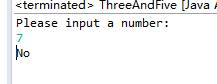
2. 编写程序时，首先使用了if语句判断给定的数能否同时被3和5整除。

3.在调试过程中，发现逻辑错误，判断条件应该使用逻辑运算符&&，而不是||。

4.修改后重新编译运行程序，得到正确的结果。

**二、实验实习结果分析**  
1、不能被3整除，但能被5整除。所以不满足要求，所以输出No，程序运行正确。  
2、能被3整除，同时也能被5整除。所以满足程序要求，输出Yes，程序运行正确。

3、被3整除，不能被5整除，不满足要求，输出No，程序运行正确。

4、不能被3整除也不能被5整除，不满足程序要求，输出No，程序运行正确。

5.测试用例包括能被3和5整除的数，以及不能被3和5整除的数。

6.覆盖了程序运行时的每个分支，验证了程序的正确性。

**三、心得体会**  
通过该实验，我学到了基本逻辑运算符的功能以及if-else选择语句结构的运用，能够成功地判断项目要求并在选择分支中完成代码编写。并且了解了新建引用和new对象的方法。我进一步加深了对逻辑运算符的理解，以及如何使用if语句进行条件判断。同时也意识到了在编写程序时需要仔细思考逻辑，避免出现逻辑错误

**题目③**

**一、源程序调试过程**

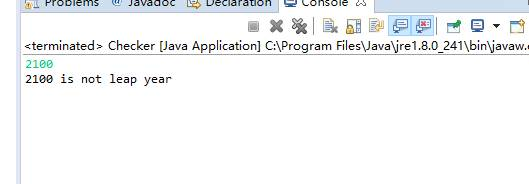
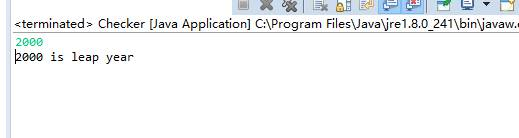
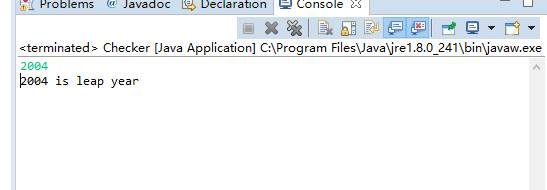
1、(year % 4 == 0 && year % 100 != 0||year%400==0)导致运行结果不对：运算符有优先级，要使用圆括号达到想要的运算顺序。应该调整为((year % 4 == 0 && year % 100 != 0)||year%400==0)  
2、在判断百年不闰时，将逻辑判断符错误写成了“！==”，应该修改为“！=”。

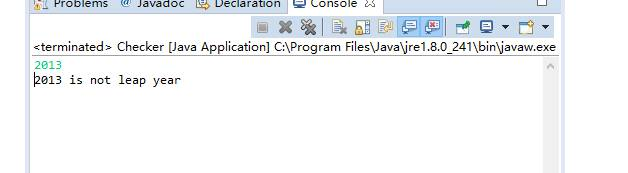
3.编写程序时，使用了if-else语句判断给定年份是否为闰年。

4.在调试过程中，注意到了闰年的判断条件，即年份能被4整除但不能被100整除，或者能

400整除.

5.编译运行程序，并使用不同的年份进行测试。

**二、实验实习结果分析**  
1、闰年百年不闰，2100是100的倍数且不是400的倍数，所以2100年不是闰年，程序运行正确。  
2、闰年400年又闰，2000是400的倍数，所以2000年是闰年，程序运行正确。  
3、闰年4年一闰百年不闰，2004是闰年，程序判断正确。

4、2013不满足任何闰年的规则，不是闰年，程序判断正确。

5. 测试用例覆盖了各种情况的年份，包括闰年和非闰年。

6.确认了程序在不同情况下的输出是否正确。

**三、心得体会**  
通过该实验，我学会了写代码时不能吝啬圆括号，多使用圆括号不仅可以避免逻辑错误，还能更易于阅读。并且强化了新建引用和new对象的方法，掌握了基本的两种初始化对象的方法。我还更加清楚了更多逻辑运算符的优先级。对if-else分支结构的运用也更加的熟练。这个实验也让我更加熟悉了Java中的条件语句的使用方法。