

SCJP035经典104题分析笔记 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/207/2021_2022_SCJP035_E7_BB_8F_c104_207276.htm 最近备考035课程，从网上当了一份据说是很有价值的104题，目前边分析边学习，已经整理出一些例题，本人才疏学浅，试发一部分笔记，不知大家是否有兴趣一起讨论。如有错误和疏忽欢迎大家指正。（示例部分未上传，如大家认为有价值我会继续发下面的笔记和示例）

```
1) public class ReturnIt{
2) returnType methodA(byte x, double y){
3) return (short)x/y*2.4 }
5) }
```

what is valid returnType for methodA in line 2? 题意：第2行中methodA方的正确返回值类型是什么？分析：这是一道考察基础知识的问题，比较简单，涉及强制类型转换和表达式数据类型的转换和升级。强制类型转换的方法是在变量或数值前加上需要转换的类型，如 (int) 12.23 则double类型的数值被强行转化为int型，当然此时回丢失信息。表达式数据类型的转换原则是在不出现信息丢失的情况下自动升级到一种更长的形式。解答：判断return后的表达式的值类型即可，(short)的强制转换仅为x，根据自动提升原则表达式的值被提升到y的类型既double,所以答案为double。注意：（1）、不带尾巴的数值容易被忽视他的真实类型，比如12和12.0，其中12为int或short型，12.0为duble型。（2）、二元操作符（如、-、*、/）当其操作的对象是基本数据类型时，会把其操作的变量自动提升为至少到int型，主要针对byte和short型。例如如下代码会出错：short a, b, c. a = 1. b = 2. c = a b.//编译指示这行出错了，possible loss of precision 示例：w01.java

```
2. 1) class Super{
2) public float
```

getNum(){return 3.0f;} 3) } 4) 5) public class Sub extends Super{ 6) 7) } which method, placed at line 6, will cause a compiler error? A. public float getNum(){return 4.0f;} B. public void getNum(){} C. public void getNum(double d){} D. public double getNum(float d){return 4.0d.}

题意：哪一种方法定义放在第6行，会产生编译错误？分析：此为一道同时考察overload和override的问题。overload是利用同一个函数名和不同的参数形式来完成不同的功能，不同的参数形式的意思是：有和无参数的区别；参数个数的区别；参数类型的区别和参数的排列方式的区别。Overload不能利用返回值来区分。Overload现象可出现在同类或父类与继承类中。override的是类继承过程中出现的现象，是对父类方法的改写，所以必须满足以下条件：1、方法名相同；2、返回类型相同；3、参数完全相同。解答：分别将答案放在第6行，A表示对super类的getNum方法进行override，正确；B语句因为参数与父类的同名函数的参数相同所以不是overload，而返回值与父类同名函数返回值不同所以不是override，进而出错。C表示对继承自super类的getNum方法进行overload，正确；D同C 注意：考察多个知识点时要综合分析，但首先要保持清醒，弄清考察的意向。示例：

```
w02.java 3. public class IfTest{ public static void main(String args[]){
int x=3. int y=1. if(x=y) System.out.println("Not equal"). else
System.out.println("Equal"). } }
```

what is the result? 题意：结果是什么？分析：也是属于基础知识的考察，请参考运算符说明部分。=为赋值，表达式本身的值为=号左边的值；==为比较，表达式本身的值为true或false。If语句后面需要一个boolean的值来判断。解答：compile error 注意：示例：w03.java 4.

`public class Foo{ public static void main(String args[]){ try{return.} finally{ System.out.println("Finally").} } }` what is the result? A. print out nothing B. print out "Finally" C. compile error 题意：结果是什么？分析：考察trycatchfinally语句的语法。一般，try语句块中产生异常，由catch语句捕获异常进行处理。关于finally语句，sun-275的教材中解释的很明确，它的代码块“总是”被执行，而不考虑有没有异常发生。不执行finally代码块的情况只有一种，就是VM结束（执行System.exit()方法和机器关闭）。如果一个return语句嵌入在catch快内，则finally快的代码在return之前执行。解答：B注意：若有System.exit()出现要注意。示例：w04.java 5.`public class Test{ public static String output="". public static void foo(int i){ try { if(i==1){ throw new Exception(). } output = "1". } catch(Exception e){ output = "2". return. } finally{ output = "3". } output = "4". } public static void main(String args[]){ foo(0). foo(1). 24 } }` what is the value of output at line 24? 题意：在第24行处，output的值是什么。分析：本题考察static类型的变量和trycatch语句的执行。trycatch语句参见第4题的讲解。所谓静态，据本人理解指的是其在内存中的位置是固定的，即程序装载完成后所有静态变量就固定于某一内存地址中，它们不随着类的实例化而随对象的地址而变动，所以有人说静态变量属于类而不属于对象。所以静态方法可以不对类进行实例化而直接调用，静态变量也可以不对其所属类进行实例化而直接使用。而多次实例化的对象使用静态变量时，实际是使用同一内存地址的内容。解答：从主程序的调用入手分析。1、foo(0)调用foo函数，并将0传给foo函数的i变量。2、运行try语句块，比较i==1不成立，执

行if后面的语句output="1"，此时output的值为“1”。3、try语句没有异常抛出，在执行finally语句后即结束。finally语句改变output的值为“13”。4、执行try语句外面的语句output="4"，此时output的值为“134”，Foo(0)调用结束。5、Foo(1)调用Foo函数并将1传给Foo函数的i变量。6、运行try语句块，比较i==1成立，执行if语句块内的语句。7、抛出一个异常。8、catch语句立刻捕获异常并执行output="2"，此时output的值为“1342”。9、继续向下执行return准备结束Foo函数，但是return前要执行finally语句的内容output="3"，此时output的值为“13423”。然后结束Foo函数。在第24行处output的值为“13423”。注意：答案时应该写“ ”？示例：
： w05.java6. public class IfElse{ public static void main(String args[]){ if(odd(5)) System.out.println("odd"). else System.out.println("even"). } public static int odd(int x){return x%2.} } what is output? 题意：输出是什么？分析：本题比较简单，只是用odd这样的函数名迷惑视线，其实odd函数定义的类型是int，所以放在if里判断自然会产生错误。解答：
complime error注意：if为常考内容（仅仅从模拟考试题得出的结论，不代表实际考试情况）100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com