1，下列说法正确的是（ ）

A.一个文件里可以同时存在两个 public 修饰的类

B.构造函数可以被重写（override）

C.子类不能访问父类非 public 和 protected 修饰的属性

D.final 修饰的类可以被继承

答案：C

一个 Java 源文件中最多只能有一个 public 类，当有一个 public 类时，源文件名必须与之一致，否则无法编译，如果源文件中没有一个 public 类，则文件名与类中没有一致性要求。至于 main()不是必须要放在 public 类中才能运行程序。

重写是子类继承父类后对父类的方法进行修改。方法名，参数，返回值必须一样。不能重写被标示为 final 的方法。如果不能继承一个方法，则不能重写这个方法。

扩展：重写 override，重载 overload 的区别

java 的方法重载

就是在类中可以创建多个方法，它们具有相同的名字，但具有不同的参数和不同的定义。调用方法时通过传递给它们的不同参数个数和参数类型来决定具体使用哪个方法,而且返回值类型可以相同也可以不相同，这也是面向对象的多态性。

java 的方法重写

父类与子类之间的多态性，对父类的函数进行重新定义。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。在 Java 中，子类可继承父类中的方法，而不需要重新编写相同的方法。但有时子类并不想原封不动地继承父类的方法，而是想作一定的修改，这就需要采用方法的重写。方法重写又称方法覆盖。

若子类中的方法与父类中的某一方法具有相同的方法名、返回类型和参数表，则新方法将覆盖原有的方法。如需父类中原有的方法，可使用 super 关键字，该关键字引用了当前类的父类。

子类函数的访问修饰权限不能少于父类的；

重写方法只能存在于具有继承关系中，重写方法只能重写父类非私有的方法。

2，for(int x=0,y=0;(y!=0)&&(x<4);x++)循环的执行次数是（ ）

A.无限次

B.执行 4 次

C.执行 3 次

D.一次也不执行

答案：D

y 初始值为 0，在整个 for 循环中，y 的值不变，故判断语句中的(y!=0)不成立，故一次也不执行。

3，关于 JAVA 堆,下面说法错误的是( )

A.所有类的实例和数组都是在堆上分配内存的

B.对象所占的堆内存是由自动内存管理系统回收

C.堆内存由存活和死亡的对象，空闲碎片区组成

D.数组是分配在栈中的

答案：D

首先数组是分配在堆中的，故 D 的说法不正确。

Java 堆的结构：JVM 的堆是运行时数据区，所有类的实例和数组都是在堆上分配内存。它在 JVM 启动的时候被创建。对象所占的堆内存是由自动内存管理系统也就是垃圾收集器回收。堆内存是由存活和死亡的对象组成的。存活的对象是应用可以访问的，不会被垃圾回收。死亡的对象是应用不可访问尚且还没有被垃圾收集器回收掉的对象。一直到垃圾收集器把这些对象回收掉之前，他们会一直占据堆内存空间。

4，在使用 super 和 this 关键字时，以下描述正确的是（ ）

A.在子类构造方法中使用 super（ ）显示调用父类的构造方法；
super（ ）必须写在子类构造方法的第一行，否则编译不通过

B.super（ ）和 this（ ）不一定要放在构造方法内第一行

C.this（ ）和 super（ ）可以同时出现在一个构造函数中

D.this（ ）和 super（ ）可以在 static 环境中使用，包括 static 方法和 static 语句块

答案：A

Java 关键字 this 只能用于方法方法体内。当一个对象创建后，Java 虚拟机（JVM）就会给这个对象分配一个引用自身的指针，这个指针的名字就是 this。因此，this 只能在类中的非静态方法中使用，静态方法和静态的代码块中绝对不能出现 this。

super 关键和 this 作用类似，是被屏蔽的成员变量或者成员方法或变为可见，或者说用来引用被屏蔽的成员变量和成员成员方法。

不过 super 是用在子类中，目的是访问直接父类中被屏蔽的成员，注意是直接父类（就是类之上最近的超类）

5，下列语句哪一个正确（ ）

A．Java 程序经编译后会产生 machine code

B．Java 程序经编译后会产生 byte code

C．Java 程序经编译后会产生 DLL

D．以上都不正确

答案：B

Java 字节码是 Java 源文件编译产生的中间文件

java 虚拟机是可运行 java 字节码的假想计算机 java 的跨平台性也是相对与其他编程语言而言的 。

先介绍一下 c 语言的编译过程：c 的文件经过 C 编译程序编译后生成 windows 可执行文件 exe 文件然后在 windows 中执行。

再介绍 java 的编译过程：java 的文件由 java 编译程序将 java 字节码文件就是 class 文件在 java 虚拟机中执行。机器码是由 CPU 来执行的；Java 编译后是字节码。

电脑只能运行机器码。Java 在运行的时候把字节码变成机器码。C/C++在编译的时候直接编译成机器码

6，下列哪一种叙述是正确的（ ）

A．abstract 修饰符可修饰字段、方法和类

B．抽象方法的 body 部分必须用一对大括号{ }包住

C．声明抽象方法，大括号可有可无

D．声明抽象方法不可写出大括号

答案：D

abstract 修饰符用来修饰类和成员方法

用 abstract 修饰的类表示抽象类，抽象类位于继承树的抽象层，抽象类不能被实例化。

用 abstract 修饰的方法表示抽象方法,抽象方法没有方法体。抽象方法用来描述系统具有什么功能，但不提供具体的实现。

Abstract 是 Java 中的一个重要关键字，可以用来修饰一个类或者一个方法。

修饰一个方法时，表示该方法只有特征签名（signature），没有具体实现，而是把具体实现留给继承该类的子类，所以不能有大括号。

7，下列说法正确的有（ ）

A．class 中的 constructor 不可省略

B．constructor 必须与 class 同名，但方法不能与 class 同名

C．constructor 在一个对象被 new 时执行

D．一个 class 只能定义一个 constructor

答案：C

这里可能会有误区，其实普通的类方法是可以和类名同名的，和构造方法唯一的

区分就是，构造方法没有返回值。

8，GC 线程是否为守护线程（ ）

答案：是

线程分为守护线程和非守护线程（即用户线程）。

只要当前 JVM 实例中尚存在任何一个非守护线程没有结束，守护线程就全部工作；只有当最后一个非守护线程结束时，守护线程随着 JVM 一同结束工作。

守护线程最典型的应用就是 GC (垃圾回收器)

9，关于 sleep()和 wait()，以下描述错误的一项是（ ）

A. sleep 是线程类（Thread）的方法，wait 是 Object 类的方法；

B. sleep 不释放对象锁，wait 放弃对象锁；

C. sleep 暂停线程、但监控状态仍然保持，结束后会自动恢复；

D. wait 后进入等待锁定池，只有针对此对象发出 notify 方法后获得对象锁进入运行状态。

答案：D

sleep 是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用 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，进入等待此对象的等待锁定池，只有针对此对象发出 notify 方法（或 notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

10，方法 resume()负责恢复哪些线程的执行（ ）

A，通过调用 stop()方法而停止的线程。

B，通过调用 sleep()方法而停止的线程。

C，通过调用 wait()方法而停止的线程。

D，通过调用 suspend()方法而停止的线程。

答案：D

suspend 可以挂起一个线程，就是把这个线程暂停了，它占着资源，但不运行，用 resume 是恢复挂起的线程，让这个线程继续执行下去。