## 一、Java语言基础

### 1.1 Java语言特性、虚拟机

**1、**下列关于Java语言的特点，描述错误的是：（ ）

A．Java是跨平台的编程语言 B．Java支持分布式计算

C．Java是面向过程的编程语言 D．Java支持多线程

（单选 Java语言特性）

**2、**JAVA平台的无关性是通过（ ）实现的   
 A．JAVA虚拟机 B．JAVA的编辑环境   
 C．Windows操作系统 D．UNIX操作系统

（单选 虚拟机）

**3、**Java语言具有许多优点和特点，下列选项中，哪个反映了Java程序并行机制的特点：（）

 A. 安全性　　B. 多线性　　C．跨平台　　D．可移植

（单选 Java语言特性）

**4、**在Java语言中，不允许使用指针体现出的Java特性是：（ ）

　 A. 可移植 B. 解释执行 C. 健壮性 D. 安全性

（单选 Java语言特性）

### 1.2 关键字、变量、标识符、数据类型

**1、(单选 字符常量类型)**下面的语句哪一行在编译时没有警告和编译错误：（ ）

A．float f=1.3;

B．char c="a";

C．byte b=257;

D．double i = 10;

解析：注意字面常量的类型，1：int 1.0:double ‘1’: char “1”:字符串

因为1.3的类型是double，所以A错

因为”a”的类型是Sring，所以B错

因为 257已超出byte所能表示的数的范围，所以C错。D正确。10可以自动提升为double类型。

答案：D

**2、(多选 float)**下面哪些是float的有效声明：（ ）   
A. float foo=-1; B. float foo=2.02f;

C. float foo=3.03d; D. float foo=0x0123;

解析：整数可以自动提升为float类型，所以A正确，2.02f是将double类型强制转换为float类型，所以B正确，3.03d与3.03一样，都是double类型，double类型不能自动转换为float类型，C错，0x123是整数的8进制表示，D正确

答案：ABD

**3、（单选 关键字）**以下哪一项不是Java的关键字：（ ）

A．then B．goto C．case D．while

解析：

Java 关键字，共50个=已用的48个+2个未用的）

基本数据类型（8）：boolean byte char double float int long short

引用类型（3）：class enum interface

语句（11）：if else for while do switch case default break continue return

修饰符(11)：abstract final private protected public static native volatile transient synchronized strictfp

异常处理(5)：try catch finally throw throws

其它(10)： void extends implements new instanceof this super import package assert

保留字（当前没有使用的关键字）：const goto。

特殊标识符，等同于字面常量（不作为关键字）：true false null

注意c语言中的关键字then sizeof不是Java的关键字或保留字

答案：A

**4、（单选 关键字）**下面哪个不是Java的关键字或者保留字：（ ）

A．if

B．then

C．goto

D．while

答案：B

**5、（单选 关键字）**哪些不是Java关键字：（ ）

A．sizeof  
 B. const  
 C. super  
 D. void

答案：A

**6、（多选 关键字）**以下选项中，有哪些是Java关键字：（ ）

A．default  B．null  C．String  D．throws  E．long

答案：ADE

**7、（多选 关键字）**以下哪些是java的关键字：（ ）

A． run B． default C． implement D． import

答案：BD

**8、(单选 标识符)**下面四个变量声明语句中,哪一个变量的命名是正确的：（ ）

A. Object while B.Object my\_house C.Object my dog D. Object 2cats

解析：合法标识符是：

1、以字母、下划线（\_）或$开头，后面可以是字母、数字、下划线、$符

2、不能是Java关键字、及特殊常量true、false及null。

本题中A使用了关键字while，C标识符有特殊字符空格，D以数字开头，这几个均错。B正确

答案：B

**9、（多选 标识符）**以下有效的标识符是？

A． IDoLikeTheLongNameClass B． $byte C． const D． \_ok E． 3\_case

答案：ABD

**10、（多选 标识符）**在Java中，下列（ ）是合法的标识符。

A．A123$ B．#A123 C．588TTT D．s123\_Name

答案：AD

**11、（多选 标识符）**下面几个变量中，那些是对的？那些是错的？错的请说明理由

A. ILoveJava

B. $20

C. learn@java

D. antony.lee

E. Hello\_World

F. 2tigers

答案：ABE

**12、（单选 局部变量初始化）**阅读以下代码：

1) import java.io.\*;

2) import java.util.\*;

3) public class foo{

4) public static void main (String[] args){

5) String s;

6) System.out.println("s=" + s);

7) }

8) }

输出结果应该是：（ ）

A．代码得到编译，并输出“s=”

B．代码得到编译，并输出“s=null”

C．由于String s没有初始化，代码不能编译通过

D．代码得到编译，但捕获到 NullPointException异常

解析：局部变量在使用前必须初始化，本题局部变量s在使用前没有初始化，故编译错误

答案：C

**13、（单选 整数的不同进制的表示）**十进制数16的16进制表示格式是\_\_\_\_。

A．0x10 B．0x16 C．0xA D．016

解析：以0x开头的数为16进制表示，以0开头的数为8进制表示，

答案：A

### 1.3 运算符

**运算符**

**1、（单选 运算符）**下列运算符合法的是：（ ）   
A、&& B、<> C、if D、:=

答案：A

**2、（单选 ++）**给定某java程序的main方法，如下

1）public static void main(String args[])

2）{

3） int i=0;

4） System.out.println(i++);

5）}

运行结果是：（ ）

A．0 B．1 C．编译错误 D．运行时出错异常

解析：注意前++与后++的区别。前++,先加变量的值,表达式为加一之后的变量的值

后++,先取变量的值作为表达式的值,再将变量加1

前++：先运算，再取值 后++：先取值，再运算

**测试代码：**

int i = 10;

int j = ++i;

int k = i++;

//i=?,j=?,k=?

System.out.println("i="+i+",j="+j+",k="+k);

本题是后++，i的值作为表达式的值，之后，i再加1

答案： A

**3、（单选 ++）**以下程序的运行结果是：（ ）

1）public class Increment{

2） public static void main(String args[]){

3） int c;

4） c = 2;

5） System.out.print(c);

6） System.out.print(c++);

7） System.out.println(c);

8） }

9）}

A．222 B．233 C．223 D．344

答案： C

（单选： ++运算符）

**4、**下面哪一行代码不是合法的赋值语句：（ ）

A. float f=1/3 B. int i=1/3 C. float f=1\*3.0 D. double d=1.0/3

解析：这是典型的二元数值类型自动提升。二元数值运算时，自动将短的数据类型提升为长的类型，表达式的类型为长的类型

Byte short char -- > int - > long ->float –> double

本题中，1/3的类型为int，1.0/3的类型为double，所以ABD均正确，1\*3.0的类型为double，double是不可以自动转换为float，故C错。

类似的面试题有：

下列代码有没有问题？

short s = 1;

s = s + 1;

注意byte、short、char之间不会自动提升，而是直接提升为int，所以

short s1 = 1;

short s2 = 2;

s1+s2的类型为int

答案： C

**5、（单选 ？：运算符）**阅读下列代码

1）public class Test2005{

2） public static void main(String args[]){

3) System.out.println((3>2)？4：5)；

4) }

5)}

其运行结果是

A、2 B、3 C、4 D、5

（单选 ？：运算符）

答案： C

**6、（单选 ？：运算符）**执行下列代码段后，

int x=9;

int y=9;

x=(y<=x++)?1:0;

变量x的值是：（ ）

A．1 B．2 C．0 D．10

答案： A

（单选 ++运算符 ？：运算符）

**7、（\*单选 ？：运算符）**编译运行以下程序后，关于输出结果的说明正确的是（）

1） public class Conditional{

2） public static void main(String args[ ]){

3） int x=4;

4） System.out.println(“value is “+ ((x>4) ? 99.9 :9));

5） }

6）}

A．输出结果为：value is 99.99 B．输出结果为：value is 9

C．输出结果为：value is 9.0 D．编译错误

解析：

三元运算符：注意表达式的类型由:前后的类型确定，如果是数值，则也会自动提升，如果是对象，则应能自动转换，表达式的类型为父类型，如：

String s = true?"java":new Object();

错误，问号表达式的类型为Object

这样写就成：

Object o = true?"java":new Object();

上面?:表达式的类型为Object

本题中，99.9:9的类型提升为double，故输出为9.0

（\*单选 ？：运算符及数值类型自动提升）

答案： C

**8、（\*单选 &&与&运算符）**给出下面的代码片断

1) String str = null;  
2) if ((str != null) && (str.length() > 10)) {  
3) System.out.println("more than 10");  
4) }   
5) else if ((str != null) & (str.length() < 5)) {  
6) System.out.println("less than 5");  
7) }  
8) else { System.out.println("end"); }

哪些行将导致错误：（ ）

A. 第1行

B. 第2行  
C. 第5行  
D. 第8行

解析：注意&&与&的区别，&&是短路运算符，&不是，而&还可以作数值运算符

本题中注意str = null，所以 str != null为false，对&&运算符，不会运算右边的表达式，故2行不错，而对&仍然会运算右边表达式，在null对象上调用方法，则会抛出NullPointerException异常。故第5行会有运行时错误。

答案： C

**9、(单选 &运算符)**阅读下列代码

1）public class Test2005{

2） public static void main(String args[]){

3） System.out.println((0xa5)&0xaa)；

4） }

5）}

其运行结果是：（ ）

A）0xa5 B）0xa0 C）0x50 D）0xaa

解析：&可作为位运算符，其含义为”与”，当对应的位均为1时，结果为1，否则为0

运算如下：

1010 0101

1010 1010

& 1010 0000

结果为：0xa0,0x是数的十六进制表示

答案： B

**10、(单选 移位运算符)**阅读下列代码

1）public class Test{

2） public static void main(String args[]){

3） System.out.println(89>>1)

4） }

5）}

其运行结果是：（ ）

　 A、44 B、45 C、88 D、90

（单选 >>运算符）

解析：往右移一位相当于除以2，这是效率最高的运算。注意>>与>>>的区别：

都是向右移位，>>最高位不变，而>>>最高位补0

测试代码：

int i = -10;

int j = i >> 1; 最高位仍然是1，所以仍然为负数

int k = i >>> 1; 最高位的1向右移动了一位，而最高位变为0，故成了一个很大的正数

//j和 k是正数还是负数?

System.*out*.println("j="+j+",k="+k);

答案： A

**11、** **(单选 运算符)**下面哪两个值相等：（多选 运算符）  
 A、3/2  
 B、3<2  
 C、3\*4  
 D、3<<2

解析，往左移一位相当于将数\*2，移2位相当于\*4

答案： CD

### 1.4 语句

**1：(多选 if语句)**以下程序片断，能通过编译的是：

A、 int i = 0;

if(i) System.*out*.println("hello");

B、 boolean b1 = true;

boolean b2 = true;

if(b1==b2) System.*out*.println("So true");

C、 int i=1;

int j=2;

if(i==1 || j==i) System.*out*.println("OK");

D、 int i=1;

int j=2;

if(i==1 &| j==i) System.*out*.println("OK");

解析：if()中的表达式结果只能是true或false,Java中没有运算符&|

答案：BC

**2、(单选 if-else运行流程)**给定下列代码:  ****

1）if(x>4){ 

2） System.out.println("Test 1");

3）}else if (x>9){ 

4） System.out.println("Test 2");

5）}else { 

6） System.out.println("Test 3");

7）} 

要使输出为"Test 2"，则x的范围是（）： 

A．x<4 B．x>4 C．x>9 D．None

解析：本题注意如果x>9，也一定x>4，会输出Test 1,如果x<4，则会输出Test 3，所以永远不会输出Test2

答案：D

**3、(单选 switch)以下哪些代码是合法的：**

A、 float x = 1;

switch(x) {

case 1:

System.out.println("Get a 1");

}

B、 long y = 1;

switch(y) {

case 1:

System.out.println("Get a 1");

}

C、 byte b = 1;

switch(b) {

case 1:

System.out.println("Get a 1");

break;

}

D、 int i = 1;

int j = 2;

switch(i) {

case j:

System.out.println("Get a 1");

break;

}

E、 short s = 1;

switch(s) {

case 1.2:

System.out.println("Get a 1");

break;

}

F、 short s = 1;

switch(s) {

case 1,2,3:

System.out.println("Get a 1");

break;

}

解析：switch-case语句中的几个语法点，

1、switch()中的表达式只能是int类型或能提升为int类型的变量即byte、short、char，而不能是其它类型，long、字符串都不支持，所以AB均错，C正确。

2、case后面必须是一个int类型的常量，不能是变量，DE错。E错，java中不支持这种语法格式。

答案：C

**4、(单选 switch)**试图编译运行下面的代码会输出什么样的结果：（ ）

1） int i=1;

2） switch (i) {

3） case 0:

4） System.out.println("zero");

5） break;

6） case 1:

7） System.out.println("one");

9） case 2:

10） System.out.println("two");

11）default:

12） System.out.println("default");

13） }

A．one

B．one, default

C．one, two, default

D．default

解析：switch的运行流程，当匹配到与case中常量相等的值，执行case中的语句，直到碰到break语句，或switch语句结束。本例中，case 1与i的值相匹配，所以执行case 1中的语句块，由于case 1中无break语句，接着执行case 2中的语句块，接着执行default中的语句块，直至switch语句结束。

答案：C

**5、(多选 switch)**给出下面的代码片断:   
1) switch(x){   
2) case 1:System．out．println(“Test 1”);break;   
3) case 2:   
4) case 3:System．out．println(“Test 2”);break;   
5) default:System．out．println(“end”);   
6) }   
which value of x would cause “Test 2” to the output:   
A．1 B．2 C．3 D．default

答案：BC

**6、(单选 switch)**下列语句中，属于多分支语句的是：（ ）

　 A. if语句 B. switch语句 C. do while语句 D. for语句

答案：B

**7、(单选 switch)**阅读下列代码

1）public class Test2009{

2） public static void main(String args[]){

3） String s=″Test″；

4） switch(s){

5） case″Java″：System.out.print(″Java″)；

6） break；

7） case″Language″：System.out.print(″Language″)；

8） break；

9） case″Test″：System.out.print(″Test″)；

10） break；

11） }

12） }

13）}

其运行结果是：（ ）

　A. Java B. Language C. Test D. 编译出错

答案：D

**8、(单选 switch)**.代码如下:   
 1）int i=1, j=0；   
 2）switch(i) {   
 3）case 2:   
 4）j+=6;   
 5）case 4:   
 6）j+=1;   
 7）case 1:   
 8）j +=2;   
 9）case 0:   
 10）j +=4;   
 11）}   
最终j的值是什么？   
A. 0 B. 2 C. 4 D. 6 E. 9   
答案：D

**1、(单选 循环语句)**.代码如下:   
 1）public class WhileExm {   
 2）public static void main (String []args) {   
 3）int x= 1, y = 6;   
 4）while (y--) {x--;}   
 5）system.out.printIn(“x=” + x “y =” + y);   
 6）}   
 7）}   
结果是什么?   
A. The output is x = 6 y = 0   
B. The output is x = 7 y = 0   
C. The output is x = 6 y = -1   
D. The output is x = 7 y = -1   
E. 编译失败

解析：注意if else if while中的表达式，必须是boolean，即运算结果，只能是true或false，所以本题会编译错误，当然如将while中的y—改为 y-->0 又是另一题了

答案：E

**2、(单选 循环语句)**.执行如下程序代码   
　　1）a=0;c=0;  
　　2）do{  
　　3） --c;  
　　4） a=a-1;  
　　5）}while(a>0);  
　　C的值是：（ ）  
　　A、0 B、1 C、-1 D、死循环

解析：循环体只执行了一次

答案：C

**3、(单选 循环语句)**以下选项中循环结构合法的是：（ ）

A. while (int i<7)

{ i++;

System.out.println(“i is “+i);

}

B. int j=3;

while(j)

{

System.out.println(“ j is “+j);

}

C. int j=0;

for(int k=0; j + k !=10; j++,k++)

{

System.out.println(“ j is “+ j + “k is”+ k);

}

D. int j=0;

do{

System.out.println( “j is “+j++);

if (j = = 3) {continue loop;}

}while (j<10);

解析：A、B中的while表达式只能是逻辑表达式，所以AB错，D中没有定义标签loop

答案：C

**4、(\*单选 方法传参)**给出下面的代码

1）public class Test {  
2）　 void printValue(int m){  
3） do { System.out.println("The value is"+m);  
4） }  
5） while( --m > 10 )  
6） }  
7） public static void main(String arg[]) {  
8） int i=10;  
9） Test t= new Test();  
10） t.printValue(i);  
11） }  
12）}

输出将是什么：（ ）

A. 8   
B. 9  
C. 10  
D. 11

解析：本题有较大的迷惑性，看上去是计算循环，实际上是方法调用，java中参数按值传参，所以i的值不会改变，仍然为10

答案：C

**5、(\*\*\*综合 for及方法调用)**运行下列程序，其输出结果为何？（ ）

1） public class Test {  
2）　　static boolean foo(char c) {  
3） 　 System．out．print(c);  
4） 　 return true;  
5） }  
6） 　public static void main( String[] argv ) {  
7） 　　int i =0;  
8） 　　for ( foo('A'); foo('B')&&(i<2); foo('C')){  
9） 　　　i++ ;  
10） 　　 foo('D');  
12） 　 }  
13） 　}  
14） }

A．ABDCBDCB B．ABCDABCD

C．Compilation fails D．An exception is thrown at runtime．

解析：算是综合了，据说是华为面试题，要准确理解方法调用及for语句的运行流程

答案：A

### 1.5 其它

**1、**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

（单选，基础，其它）

**2、**JAVA原程序经过编译器编译后生成的字节码文件的后缀是（ ）   
A．java B．c   
C．class D．src

（单选，基础，其它）

**3、**JDK中提供的文档生成器是：（ ）

A. java.exe B. javap.exe C. javadoc.exe D. javaprof.exe

（单选，基础，其它）

**4、**Java语言使用的字符码集是：（ ）

　 A. ASCII B. BCD C. DCB D. Unicode

（单选，基础，其它）

### 1.6 命令行参数，Main方法

**1、**main()方法的返回类型是?

A．int B．void C．Boolean D．static (main方法)

（单选，main方法）

**2、（命令行参数及数组）**下面的代码在输入下面的命令行后会输出什么内容：（ ）

命令行：java myprog good morning

代码：

1）public class myprog{

  2） public static void main(String argv[])

  3）  {

  4）      System.out.println(argv[2]);

  5）   }

6）}

A．myprog

B．good

C．morning

D．Exception raised: "java.lang.ArrayIndexOutOfBoundsException: 2"

（单选，命令行参数及数组）

**3、**java中程序的main方法必须声明为（ ） main方法  
A．public B．void   
C．public void D．public static void

（单选，main方法）

## 二、数组

1. **（多选 定义数组）**以下哪些是合法的数组声明和初始化？

A、int x[] = {1,2,3}; B、int[]x[] = {{1,2,3},{1,2,3}};

C、int x[3] = {1,2,3}; D、int[] x = {0,0,0};

E、char c[] = {‘a’,’b’}; F、int x[] = new int[]{1,2,3};

G、int x[] = new int[3]{1,2,3}; H、int[][] x = new int[][2];

I、int x[] = new int[-4]; J、int[] x = new int[0];

解析：数组定义的几种方式

一维数组：int[] arr = {1,2,3}; 定义了数组同时初始化了元素

[]可放在类型int的后面，也可放在变量arr的后面，

int[] arr = new int[]{1,2,3}; 定义了数组同时初始化了元素，注意此中情形不能有长度的定义，所以new int[len]{1,2,3}错误的

int[] arr = new int[6]; 定义了数组，但没有初始化，这时必须定义长度

ADEF定义并初始化了数组元素

B定义并初始化了一个二维数组，声明时[]可放在变量前，也可放在变量后，Java习惯放在变量前

C错，声明时无法确定数组元素个数

G错，若已初始化了元素，则不可定义其长度。

H错，定义多维数组时，必须定义数组的长度，即定义第一个[]中的数字

I：编译时没错，但运行时错

J：正确，没有元素的数组

答案：ABDEFJ

2、**（多选 定义数组）**以下哪些是合法的二维数组声明和初始化？

A、int x[][] = new int[10,10]; B、int x[][] = new int[10][10];

C、int x[10][10] = new int[10][10]; D、int[][] x = new int[10][10];

解析：A错，Java中没有[10,10]这种语法，C错，声明时不定义数组元素个数

答案：BD

3、**（单选 数组长度）**如何获得数组arr的长度？

A、arr.length B、arr.size C、arr.length() D、arr.size()

解析：Java中数组的长度通过其属性length获取，注意不是方法

答案：A

4、**（单选 数组长度）**给出下面声明，假设已经初始化了该数组，则下面哪个表达式会返回数组的大小。

int[][] array = new int[5][];

请选出一个正确的答案。

A、array[].length();

B、array.length();

C、array[2].length;

D、array[0][0].length;

E、array[3].size();

F、array.size();

解析：数组对象只有length属性，没有length方法，也没有size方法，所以ABEF均错，array是一个二维数组，array[0][0]取到的是一个元素，array[2].length是array[2]的长度

答案：C

**5、（访问数组）**以下三组代码能否编译通过？能否正常运行？

A、int[] iarr = new int[5];

System.out.println(iarr[0]);

B、 Circle[] circles = new Circle [5];

System.out.println(circles [0].toString());

C、 int[] iarr = new int[5];

System.out.println(iarr[5]);

解析：当数组没有初始化时，系统会为数组元素清0，所以A中的iarr[0]会输出为0，而B中的circles[0]为null，调用其方法时，在运行时会抛出NullPointerException，Java在编译时不检查数组索引的合法性，iarr[5]的5已越界，运行时抛出ArrayIndexOutofBoundsException异常。

答案：均能编译通过，C运行正常，

**6、（单选 数组）**执行完以下代码int [ ] x = new int[10]；后，以下哪项说明是正确的：（ ）

A．x[9]为0 B．x[9]未定义 C．x[10]为0 D．x[0]为空

解析：数组的索引从0开始，所以x[10]的索引已越界，创建数组时，如没有初始化，则已有默认值，如BD错。

答案：A

**7、（多选 数组）**下面选项正确声明一个一维数组的有：（ ）

A. String [] a B. String a[] C. char a[][] D. String a[10]

解析：数组声明的[]可放在类型的后面也可以放在变量名的后面，所以AB正确，C是二维数组的声明，在声明数组时不可以定义长度，故D错。

答案：AB

**8、（单选 数组）**试图编译运行下面的代码会发生什么情况：（ ）

1）public class Q {

2）  public static void main(String argv[]){

3） int anar[]=new int[]{1,2,3};

4） System.out.println(anar[1]);

5） }

6）}

A．1

B．Error anar 被引用前没有初始化

C．2

D．Error: 数组大小没有定义

解析：定义数组时初始化了该数组，所以anar[1]的值为2

答案：C

## 四、面向对象基础

### 4.1、类的定义与对象的创建

**2、（多选 new运算符）**给出下面的代码片断:

1) public void create() {

2)   Vector myVect;  
3)   myVect = new Vector();            
4) }

下面的哪些陈述是正确的?

A．The declaration on line 2 does not allocate memory space for the   variable myVect．

B．The declaration on line 2 allocates memory space for a reference to a Vector object．

C．The statement on line 2 creates an object of class Vector．

D．The statement on line 3 creates an object of class Vector．

E．The statement on line 3 allocates memory space for an object of  class Vector

（ADE）

### 4.2、访问控制符

private成员只能在类中可见，其它类（含其子类）均不可见。

包级成员只能被同一包中的其它类可见，其它类（含其子类）均不可见

protected成员能被同一包中的其它类及其子类可见，不在同一包中的其它类不可见

public成员能被其它任一类可见

1、类的设计要求它的某个成员变量不能被外部类直接访问。应该使用下面的哪些修饰符获得需要的访问控制：（ ）

A. public  
B. 不加修饰符  
C. protected  
D. private

（单选 访问控制符）

**1、（\*继承、包级访问修饰符）**

有如下代码

1) //MyClass.java

2) package corejava.ch6;

3) public class MyClass{

4) int value;

5) }

6)

7) //MySubClass.java,注意MySubClass与MyClass不在同一个包中

8) public class MySubClass extends MyClass{

9) public MySubClass(int value){

10) this.value = value; //this.value访问不到父类的包级成员,error

11) }

12) }

选择正确答案：（E）

A. 编译通过

B. 编译不通过，应把第10 行改成super.value = value;

C. 编译不通过，应把第10 行改成super(value);//父类没有带参数的构造方法

D. 编译不通过，可以为MySubClass 增加一个value 属性

//也可以但隐藏了父类的value

E. 编译不通过，把第四行改为protected int value; 把第10 行改为super.value = value;

//不在同一个包中的类，不能访问私有及包级变量与方法,改为protected则OK

//若子类与父类不在同一个包中，子类不能访问父类的包级成员

**1、**在Java中，下列哪句是正确的:（ ）  
A．子类的构造方法与父类的构造方法同名   
B．子类可以继承父类的所有方法 私有的东西不被继承  
C．子类可以继承父类protected和public的方法   
D．子类可以继承父类protected和private的方法

（单选 继承时的构造方法）

**2、（\*继承、protected访问修饰符）**

有如下代码

//MySuperClass.java

package corejava.chp6;

class MySuperClass{

protected int value;

}

//TestMain.java，注意MySubClass与MySuperClass不在同一个包中

import corejava.chp6.\*;

class MySubClass extends MySuperClass{

public void print(){

System.out.println(value);

}

}

//TestMain与MySubClass在同一个包中

public class TestMain{

public static void main(String args[]){

MySubClass msc = new MySubClass();

msc.print();

System.out.println(msc.value); //不可见

}

}

这段代码能否编译通过？如果可以，输出结果是什么？如果不能，原因是什么？

//编译不通过，protected仅在子类中可见，其他不在同一包中的类中不可见

**3、（\*多选 继承 访问控制 静态方法访问实例变量）**给出下面的代码片断:

1）class Person {  
2）  private int a;  
3） public int change(int m){ return m; }  
4）}     
5) public class Teacher extends Person {  
6   public int b;  
7) public static void main(String arg[]){  
8）  Person p = new Person();

9) Teacher t = new Teacher();  
10) int i;  
     // point x        
11) }  
12) }  
在// point x处的哪些申明是句法上合法的:

A．i = m;  
B．i = b;  
C．i = p．a;      
D．i = p．change(30);   
E．i = t．b;

（DE）

### 4.3、构造方法

要点：

构造方法用于初始化对象的成员变量，用在new运算符之后

构造方法的方法名与类名相同，没有返回类型

构造方法可以重载

默认构造方法：若类中没有定义构造方法，则编译器自动生成一个不带参数的构造方法，该方法体为空，若在类中已定义了构造方法，则编译器不会生成默认的构造方法。

构造方法不被继承

构造方法中可使用this/()调用其它构造方法，this调用必须是构造方法中的第一条语句

子类的构造方法中的第一条语句总是super()语句，来调用父类的构造方法，如果没有显式的使用，则默认为super（），即调用父类的无参构造方法

若子类中既有super又有this，则先super，再this

**1、（单选 构造方法）**下列关于构造方法说法正确的是（ ）   
A．构造方法必须是和类的名称相同   
B．每一个类都必须显式声明自己的构造方法   
C．构造方法不能进行重载   
D．子类不能使用父类的构造方法

解析：构造方法名与类名必须相同，A对，类可以不显式的声明构造方法，此时编译器自动生成不带参数的构造方法，B错，构造方法可以重载，C错，子类可以通过super调用父类的构造方法，D错

答案：A

**2、（多选 构造方法重载）**读以下程序：

1）public class ConstOver{   
2）　public ConstOver(int x, int y, int z){}   
3）}   
下面哪两个是构造方法的重载：（ ）  
 A. ConstOver(){}   
 B. protected int ConstOver(){}   
 C. private ConstOver(int z, int y, byte x){}   
 D. public void ConstOver(byte x, byte y, byte z){}

解析：构造方法重载现一般方法重载规则相同，即通过方法参数个数或参数类型以区分不同的构造方法

答案：AC

**3、（多选 构造方法）**关于构造方法，下列说法正确的是：

A. 每个类中都有至少一个构造方法

B. 一个类中可以有多个构造方法

C. 构造方法可以有返回值

D. 构造方法可以有多个参数

答案：ABD

**4、(改错 构造方法)**有以下代码

class MyClass{

int value;

}

public class TestMyClass{

public static void main(String args[]){

MyClass mc1 = new MyClass();

MyClass mc2 = new MyClass(10);

System.out.println(mc1.value);

System.out.println(mc2.value);

}

}

问：这个程序能否编译通过？如果可以，输出结果是什么？如果不可以，则应该如何修改？

解析：不可以，类中没有定义MyClass(int x)的构造方法，在MyClass类中加上这个构造方法

**5、（\*构造函数，this 关键字）**程序改错

public class Student{

public void Student(){} //1

void init(){

age = 10;

name = "limy";

}

public Student(String name){

this.init();

this.name = name;

}

public Student(String name, int age){

this.init();//2

this(name); // 3

this.age = age;

}

int age;

String name;

}

解析：若在构造方法中使用this语句调用另一个构造方法，这个语句必须是第一条语句，所以3处代码出错，可放在2处语句之前。此外，1处定义了一个普通实例方法，而不是构造方法

**6、（\*默认构造方法 super）**有以下代码

class Super{

}

class Sub extends Super{

public Sub(){} //

public Sub(String str){

super(str);

}

}

问：该程序应该如何修改才能编译通过？

解析：子类中用super(str)显式调用父类构造方法，而父类中没有此构造方法，故编译错误，可将super(str)调用改为super()，或在父类中增加有字符串参数的构造方法

### 4.4、对象构造过程

先父类 再子类

先静态 再实例

先成员 再构造

**1、（\* 对象构造 静态初始化代码块）**有如下代码

class MyClass{

static int i = 10;

static {

i = 20;

System.out.println("In Static");

}

public MyClass(){

System.out.println("MyClass()");

}

public MyClass(int i){

System.out.println("MyClass(int)");

this.i = i;

}

}

public class TestMain{

public static void main(String args[]){//

MyClass mc1 = new MyClass(); // 1

System.out.println(mc1.i); // 2

MyClass mc2 = new MyClass(10); // 3

System.out.println(mc2.i); // 4

}

}

请写出该程序运行的结果:

解析：执行语句1时，先静态初始化，i的值为20，输出：In Static

接着执行无参构造方法，输出MyClass()

执行语句2时，注意mc1.i就是静态变量i的值，输出为20

执行语句3时，不再静态初始化，直接调用带int参数的构造方法，输出MyClass(int)，并且静态变量i的值更新为10

执行语句4，输出静态变量i的值为10

答案：In Static MyClass() 20 MyClass(int) 10

**2、（\*\*对象构造 初始化代码块）**对象创建过程

public class ObjectInitial {

int i = 20; //实例成员变量

{//实例初始化块

System.out.println(i);

}

static String s = "java";

static { //静态初始化块

System.out.println(s);

}

//

//构造函数

public ObjectInitial(int i) {

this.i = i;

}

public static void main(String[] args) {

ObjectInitial o = new ObjectInitial(10);//1

System.out.println(o.i);//2

ObjectInitial o1 = new ObjectInitial(25);//3

System.out.println(o1.i);//4

}

}

运行以上代码，其输出是什么？

解析：执行语句1时，先静态初始化，输出java，接着实例初始化，输出i的值为20,再执行构造方法，将10赋值给实例变量i

执行语句2时，实例变量i的值为10，输出为10

执行语句3时，不再静态初始化，先实例初始化，输出i的值仍为20，再执行构造方法，将25赋值给实例变量i

执行语句4时，实例变量i的值为25，输出为25

答案：Java 20 10 20 25

**3、（\*\* 对象构造 this 和super 关键字）**有以下代码

class Super{

public Super(){

System.out.println("Super()");

}

public Super(String str){

System.out.println("Super(String)");

}

}

class Sub extends Super{

public Sub(){ //调用父类的Super()

System.out.println("Sub()");

}

public Sub(int i){

this(); //仍然是先调用父类的默认构造方法

System.out.println("Sub(int)");

}

public Sub(String str){

super(str);

System.out.println("Sub(String)");

}

}

public class TestSuperSub{

public static void main(String args[]){

Sub s1 = new Sub(); //1

Sub s2 = new Sub(10); //2

Sub s3 = new Sub("hello");//3

}

}

写出该程序运行的结果。

解析：

运行语句1时，调用Sub的无参构造方法，先调用父类的无参构造方法，输出为Super()

再运行自已的构造方法，输出为Sub()

运行语句2时，调用Sub的带int参数的构造方法，该构造方法使用自已的无参构造方法初始化对象，与语句1类似先调用父类的无参构造方法，输出为Super(),再运行无参构造方法，输出Sub()

再运行自已的构造方法，输出为Sub(int)

运行语句3时，调用Sub的带字符串参数的构造方法，先调用父类的super(str)，输出为Super(String)

再运行自已的构造方法，输出为Sub(String)

答案：

//1输出: Super() Sub()

//2输出：Super() Sub() Sub(int)

//3输出: Super(String) Sub(String)

**4、（\*继承、对象构造过程）**有以下代码

class ClassA{

public ClassA(){

System.out.println("ClassA()"); //1.2.1

}

}

class ClassB{

public ClassB(){

System.out.println("ClassB()"); //1.1

}

}

class ClassC extends ClassA{

public ClassC(){

System.out.println("ClassC()");

}

}

class ClassD extends ClassB{

private ClassA ca = new ClassA(); //1.2

private ClassC cc;

public ClassD(){

System.out.println("ClassD()");

}

public ClassD(int i){

cc = new ClassC();

System.out.println("ClassD(int)");

}

}

public class TestConstructors{

public static void main(String args[]){

ClassD cd1 = new ClassD();//1

ClassD cd2 = new ClassD(10);//2

}

}

写出这段代码的输出结果。

答案：

ClassB() ClassA() ClassD()

ClassB() ClassA() ClassA() ClassC() ClassD(int)

基本原理，先初始化父类，再初始化子类，如果子类没有显式的调用父类的构造函数，则默认调用其父类的不带参数的构造函数

解析：运行语句1,使用ClassD的默认构造方法初始化，

由于ClassD继承了ClassB，调用ClassB的默认构造方法初始化ClassB对象， 故输出ClassB()

接着运行ClassD成员变量处的初始化，会new ClassA（）,调用ClassA的默认构造方法，输出ClassA()

接着运行ClassD的默认构造方法，输出为ClassD()

运行语句2：使用ClassD的带int的构造方法初始化，由于ClassD继承了ClassB，调用ClassB的默认构造方法初始化ClassB对象， 故输出ClassB()

接着运行ClassD成员变量处的初始化，会new ClassA（）,调用ClassA的默认构造方法，输出ClassA()

接着运行ClassD的带int的构造方法，其中创建ClassC对象，由于ClassC继承了ClassA，先运行ClassA的默认构造方法，输出为ClassA，再运行ClassC的默认构造方法，输出为ClassC，再运行ClassD的有int参数的构造方法，输出为ClassD(int)

**5、（\*\*\*静态初始化代码块，多个对象的创建过程）**有以下代码

class ClassA{

static {//静态初始化块

System.out.println("In ClassA Static");

}

public ClassA(){//默认构造方法

System.out.println("ClassA()");

}

}

class ClassB{

static {

System.out.println("In ClassB Static");

}

public ClassB(){

System.out.println("ClassB()");

}

}

class ClassC extends ClassB{

static{

System.out.println("In ClassC Static");

}

public ClassC(){

System.out.println("ClassC()");

}

}

class MyClass {

static ClassA ca = new ClassA(); //

ClassC cc = new ClassC();

static{

System.out.println("In Static MyClass");

}

public MyClass(){

System.out.println("MyClass()");

}

}

public class TestMain{

public static void main(String args[]){

MyClass mc1 = new MyClass();//1

MyClass mc2 = new MyClass();//2

//false，成员变量初始化，每创建一次对象，执行一次

System.out.println(mc1.cc == mc2.cc); //3

System.out.println(mc1.ca == mc2.ca); //4

}

}

写出这个程序运行的结果。

分析：首先装载类MyClass，执行MyClass的静态初始化，

创建ClassA对象，装载ClassA，运行其静态初始化块，输出In ClassA Static

接着即创建对象ClassA，并不需要等其它类装载完

执行ClassA的构造方法，输出：ClassA()

（注意，对变量的实例声明，不会装载该类，也不会初始化，）

接着执行MyClass的静态初始化块，输出：In Static MyClass

接着创建MyClass对象

执行成员变量处的初始化，

先装载ClassC，ClassC继承了ClassB，

先执行ClassB的静态初始化, ，输出In ClassB Static

执行ClassC的静态初始化，输出In ClassC Static

接着初始化ClassC对象

先初始化父类，运行ClasssB的无参构造方法，输出，ClassB()

再初始化ClassC，运行ClassC的无参构造方法，输出，ClassC()

最后初始化MyClass对象，运行MyClass无参构造方法，输出MyClass()

对mc2，所有的静态成员不再运行，创建MyClass再做一次，输出为：

ClassB() ClassC(),MyClass()

从本例可看到一个规则：类的延迟加载，有继承关系时先加载父类，类没有关系时延迟加载（用到时再/加载）

答案：

**6、（\*\*\*初始化、父类调用有重写的方法）**看下面代码，写出程序运行的结果

public class TestDemo {

int i;

public TestDemo() {

add(2);

}

void add(int v) {

i = i + 2 \* v;

}

public static void main(String[] args) {

TestDemo d = new SubTestDemo();

d.add(4); //

System.out.println(d.i);

}

}

class SubTestDemo extends TestDemo {

@Override

void add(int v) {

i = i + 4 \* v;

}

}

解析：子类重写了父类的add方法，父类构造方法调用了add方法，这个方法是父类的方法还是子类的方法由对象的实际类型确定（多态），这里实际对象是子类对象，所以调用的是子类对象上的add方法

输出结果是24，注意：父类构造方法调用的add方法是子类所重写的方法

**7、（单选 垃圾回收）**对于垃圾回收机制(Garbage Collection)的说明，下面哪个叙述是正确的? （ ）

A．程序开发者必须编写线程来负责对内存空间的释放。

B．垃圾回收机制会检查并释放那些不再需要使用的内存空间。

C．垃圾回收机制允许程序开发者直接立即释放内存空间。

D．垃圾回收机制可设定在预期的时间内释放被Java对象所使用的内存空间。

（B）

### 4.4、方法重载

同一个类中，方法名相同，参数类型或个数不同的方法为方法重载，注意：

1）方法类型不参与方法重载

2）重载时包含从父类中继承的非私有的方法

**1、（单选 方法重载）**类Test1定义如下：

1)public class Test1

2) public float aMethod（float a，float b）{}

3)

将以下哪种方法插入行3是不合法的：（ ）

A. public float aMethod（float a，float b，float c）{ }

B. public float aMethod（float c，float d）{ }

C. public int aMethod（int a，int b）{ }

D. private float aMethod（int a，int b，int c）{ }

(B)

**2、(多选 方法重载）**下面哪几个函数是public void example(){...}的重载函数：（ ）

A. public void example( int m){...}

B. public int example(){...}

C. public void example2(){...}

D. public int example ( int m, float f){...}

(AD)

**3、(\*单选 重载方法调用时的参数匹配)** 有一个类(class)如下：

1）class Test {

2）　void test(int i) {

3）　　System．out．println(“I am an int．”);

4）　}

5）　void test(String s) {

6）　　System．out．println(“I am a string．”);

7）　}

8）

9）　public static void main(String args[]) {

10）　　Test t=new Test();

11）　　char ch='y';

12）　　t．test(ch);

13）　}

14）}

针对上述Test类，以下的叙述句中，哪个是正确的? （ ）

A．Line 5 will not compile, because void methods cannot be overridden．

B．Line 12 will not compile, because there is no version of test() that takes a char argument．

C．The code will compile but will throw an exception at line 12．

D．The code will compile and produce the following output: I am an int．

（D)

**4、（重载，实例变量）**有以下代码：

class ClassA{

public void method(int value){

System.out.println(value);

}

public void method(){

System.out.println(value);

}

int value;

}

class TestClassA{

public static void main(String args[]){

ClassA classA = new ClassA();

classA.value = 10;

classA.method();

classA.method(20);

}

}

请选择正确结果：

A. 编译不通过

B. 输出10 10

C. 输出10 20

D. 输出0 20

(C)

**5、（方法重载，方法返回值）**有以下代码

class ClassA{

void method(){

System.out.println("method()");

}

int method(int i){

System.out.println("method(int)");

}

public static void main(String args[]){

ClassA a = new ClassA();

a.method();

a.method(10);

}

}

该程序是否能编译通过？如果可以，写出该程序运行结果。如果不能，请说明理由，以及如何修改。

不能编译通过，int method()方法中应有返回值，在该方法中加return i;语句

看上去是方法重载的知识点，实际上是方法的返回

**6、（多选 方法重载）**

1）public class MethodOver{

2） public void setVar(int a, int b, float c){}

3）}

which overload the setVar?

A.private void setVar(int a, float c, int b){}

B.protected void setVar(int a, int b, float c){}

C.public int setVar(int a, float c, int b){return a;}

D.public int setVar(int a, float c){return a;}

(ACD)

### 4.5、实例成员与静态成员

**知识要点：**

静态方法不能直接调用实例方法，只能通过对象调用

静态变量为所有实例所共享

**1、（static 属性）**有如下代码

class MyClass{

static int a;

int b;

}

public class TestMain{

public static void main(String args[]){

MyClass mc1 = new MyClass();

MyClass mc2 = new MyClass();

mc1.a = 100;

mc1.b = 200;

mc2.a = 300;

mc2.b = 400;

System.out.println(mc1.a);

System.out.println(mc1.b);

System.out.println(mc2.a);

System.out.println(mc2.b);

}

}

请写出程序输出结果。

300 200 300 400 此题可用来说明实例变量与静态变量的区别

**2、（多选 实例方法、静态方法访问实例成员、静态成员）**有如下代码

class MyClass {

int a;

static int b;

void fa(){}

static void fb(){}

public void m1(){

System.out.println(a); //1

System.out.println(b); //2

fa(); //3

fb(); //4

}

public static void m2(){

System.out.println(a); //5,

System.out.println(b); //6

fa(); //7

fb(); //8

}

}

请问哪些行会编译出错？

A. //1

B. //2

C. //3

D. //4

E. //5

F. //6

G. //7

H. //8

（EG）

**3、（单选 静态方法调用实例方法）**下面的代码编译后会出现什么问题：（ ）

1）public class MyClass {

2）  public static void main(String arguments[]) {

3）       amethod(arguments);

4） }

5）  public void amethod(String[] arguments) {

6）       System.out.println(arguments);

7）       System.out.println(arguments[1]);

8）  }

9）}

A．错误，不能静态引用amethod方法

B．错误，main方法不正确

C．错误，数组必须包含参数

D．method必须声明为String类型

（A）

**2、（单选 静态方法访问产例变量）**给出下面的代码

1)public class Person{  
2)　 int arr[] = new int[10];  
3)　 public static void main(String a[]) {  
4) System.out.println(arr[1]);  
5) }  
6)}

哪些叙述是对的：( )

A、编译时出错   
B、编译时正确而运行时出错  
C、输出0  
D、输出null

（A）

**3、（单选 静态变量及初始化）**试图编译运行下面的代码会发生什么情况：（ ）

1）public class MyClass{

2）  static int i;

3）  public static void main(String argv[]){

4）  System.out.println(i);

5）  }

6）}

A．错误，变量i没有被初始化

B．null

C．1

D．0

（D）

**4、(单选 静态初始化块)** 关于以下application的说明，正确的是：（ ）

1）class StaticStuff

2） {

3）     static int x=10；

4）     static { x+=5；}

5）    public static void main（String args[ ]）

6）    {

7）       System.out.println(“x=” + x);

8）    }

9）      static { x/=3;}

10）}

A、4行与9行不能通过编译，因为缺少方法名和返回类型

B、9行不能通过编译，因为只能有一个静态初始化器

C、编译通过，执行结果为：x=5

D、编译通过，执行结果为：x=3

（D）

**5、（单选 静态变量的含义）**关于以下程序代码的说明正确的是：（ ）

1）class HasStatic{

2） private static int x=100；

3） public static void main(String args[ ]){

4） HasStatic hs1=new HasStatic( );

5） hs1.x++;

6） HasStatic hs2=new HasStatic( );

7） hs2.x++;

8） hs1=new HasStatic( );

9） hs1.x++;

10）    HasStatic.x- -;

11)     System.out.println(“x=”+x);

12)   }

13) }

A、 5行不能通过编译，因为引用了私有静态变量

B、 10行不能通过编译，因为x是私有静态变量

C、 程序通过编译，输出结果为：x=103

D、 程序通过编译，输出结果为：x=102

（D）

**19、（静态属性）**有如下代码

class MyClass {

static int count = 0;

public MyClass(){

count++;

System.out.println(count);

}

}

public class TestMain{

public static void main(String args[]){

MyClass mc1 = new MyClass();

MyClass mc2 = new MyClass();

MyClass mc3 = new MyClass();

}

}

请写出该程序运行时输出的结果。

123

**21、（\*多选 static）**以下哪些论述是正确的

A. 静态方法中不能调用非静态方法 //

B. 非静态方法中不能调用静态方法

C. 静态方法不能被覆盖 //

D. 静态方法能够用类名直接调用 //

E. 可以在不产生任何一个对象的情况下调用静态方法 //

F. 静态方法里可以使用this

(ACDE)

**22、（\*static）**

public class X{

private static int a;

public static void main(String[] args){

modify(a);

System.out.println(a);

}

public static void modify(int a){

a++;

}

}

what is the result?

0 此题实际上是方法调用

## 五、面向对象高级特性-继承、多态、接口

### 5.1、继承

**1、**下列关于关键字this和super的说法正确的是（ ）   
A．this是调用父类的方法和变量   
B．super是调用本类中的方法   
C．super是调用父类的方法和变量   
D．没有区别

（单选 this super）

**2、（多选 继承接口的基本概念）**下面关于继承的叙述,正确的是:

A．在Java编程语言只允许单继承．   
B．在Java编程语言中只允许一个类实现一个接口.

C．在Java编程语言中一个类不能同时继承一个类和实现一个接口.

D．在Java编程语言中单继承能够使代码更健壮.

（AD）

**3、(单选 继承的基本概念)**下列叙述中，错误的是：( )

　 A、父类不能替代子类

　 B、子类能够替代父类

　 C、子类继承父类

　 D、父类包含子类

（D）