1.下面代码执行的结果是多少

System.out.println('a'+1); //哈希值运算

System.out.println("a"+1); //字符串的拼接

答： 98 和 a1

2. 下列代码①处和②处正确吗？为什么

byte b,b1 = 5,b2 = 2;

b = 5 + 2;① //正确 直接赋值，不涉及其他类型的运算，仅仅是基本类型的运算，就相当于赋值

b = b1 + b2;② //错误 两个byte变量相加是一个不确定的值，默认是int

3. 下列代码执行的结果是多少？

int z=0;

z=z++;

z=z++;

z=z++;

System.out.println(z);

0

4. 下面 ①和②能否编译通过为什么？

short s = 4;

s = s + 5;① 不通过 short+int 为 int 无法直接赋值给short

s += 5;② 通过 赋值运算符可以自动完成类型转换

5.下面代码执行完毕的结果是多少？

boolean b1= true?false:true==true?false:true;

boolean b2= false?true:false==false?true:false;

boolean b3= false?false:true==true?false:true;

boolean flag= b1==b2?b2:b3;

System.out.println(b1); //false

System.out.println(b2); //true

System.out.println(b3); //false

System.out.println(flag); //false

6.下列代码执行的结果是多少？

public class Test {

public static void main(String[] args)

{

int i = 0;

for (foo('A'); foo('B') && (i < 2); foo('C'))

{

i++;

foo('D');

}

}

public static boolean foo(char c)

{

System.out.print(c + " ");

return true;

}

}

A B D C B D C B

7.下列代码执行的结果是？

public class GrandFather {

public GrandFather()

{

System.out.println("1");

}

static

{

System.out.println("2");

}

{

System.out.println("3");

}

}

public class Father extends GrandFather {

public Father()

{

System.out.println("4");

}

static

{

System.out.println("5");

}

{

System.out.println("6");

}

}

public class Child extends Father {

public Child()

{

System.out.println("7");

}

static

{

System.out.println("8");

}

{

System.out.println("9");

}

}

public class TestGo {

public static void main(String[] args)

{

Child test3Child = new Child();

}

}

2

5

8

3

1

6

4

9

7

8.下列代码执行的结果是？

public class Test {

public static void main(String[] args)

{

Test t = new Test();

}

public Test()

{

System.out.println("1");

}

static

{

System.err.println("2");

}

{

System.out.println("3");

}

}

输出不唯一 2的位置不固定

9.下列代码执行的结果是？

public class Test {

public static void main(String[] args)

{

Calendar c = Calendar.getInstance();

c.set(Calendar.YEAR, 2008);

c.set(Calendar.MONTH, 1);

c.set(Calendar.DATE, 32);

SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");

System.out.println(sdf.format(c.getTime()));

}

}

2008/03/03

10.下列代码执行的结果是？

public class Test {

String str = new String("hangzhou");

char[] ch = { 'a', 'b', 'c' };

public static void main(String[] args)

{

Test t = new Test();

t.changeStr(t.str, t.ch);

System.out.print(t.str + " and ");

System.out.println(t.ch);

}

public void changeStr(String str, char ch[])

{

str = "guigu";

ch[0] = 'q';

}

}

hangzhou and qbc

11.下列代码执行的结果是？

public class Test {

public static void main(String[] args)

{

Map<String, String> map = new HashMap<String, String>();

map.put(String.valueOf(System.currentTimeMillis()) + "a", "1");

map.put(String.valueOf(System.currentTimeMillis()) + "b", "2");

map.put(String.valueOf(System.currentTimeMillis()) + "c", "3");

for (Map.Entry<String, String> entry : map.entrySet())

{

System.out.println(entry.getValue());

}

}

}

结果不确定 //

12、下列代码执行的结果是？

public class TestEq {

public static void main(String[] args) {

// boolean equals(Object anObject)

//将此字符串与指定的对象比较。

String s1="hello";

String s2="hello";

String s3= new String("hello");

String s4=new String("hello");

System.out.println(s1==s2); //引用的比较

System.out.println(s1==s3);

System.out.println(s3==s4);

System.out.println(s1.equals(s2)); //值的比较

System.out.println(s1.equals(s3));

System.out.println(s3.equals(s4));

}

}

true

false

false

true

true

true

13、下列代码执行的结果是？

importjava.util.Date;

public class Test extends Date{

public static voidmain(String[] args) {

new Test().test();

}

public void test(){

System.out.println(super.getClass().getName());

}

}

com.guigu.demodate.Test