

JAVA03 答

1)

八种数据类型分别是整型：**byte short int long**；字符型：**char**；浮点型：**float double**；布尔型：**boolean**

2)

	字节数	表示范围
byte	1	-2⁷ 到 2⁷-1
short	2	-2¹⁵ 到 2¹⁵-1
int	4	-2³¹ 到 2³¹-1
long	8	-2⁶³ 到 2⁶³-1

Ps: 看到这个范围，我想知道这个范围是如何决定的，所以在经过学习后我了解到其分别对应的位数就是二进制表示数值时的位数，那么以 **int** 为例的话，正数时，符号位为 **0**，剩下全为一，也就是从 2^0 开始到 2^{30} ($32-2$) 求和，共有 **31** 项，所以和为 $2^{31}-1$ 。负数则是符号位为 **1**，其他位全为 **0**，这样取反后加 **1** 得到的值才是 **100.....00** (共 **31** 位) 为最小，转换成十进制就是 -2^{31} 。

3)

```
int a=4
```

```
char c='0';
```

```
int b=a+c;
```

为自动类型转换，因为强制类型转换总是显性的，需要明确地使用对应语法以转换。此处 **b** 的值为 **52**，因为 **c** 经过自动转换对应的值为 **48**，加上 **4** 则为 **52**。（**char** 转化为 **int** 时，**0** 对应的 **Unicode** 值为 **48**）

Ps: 那为啥 **0** 就对应 **48** 呢，于是经过学习我又了解到了计算机字符编码标准 (**ASCII** 和 **Unicode**)，这个标准 (**ASCII**) 将 **128** 个字符对应定义为数字、字母和一些控制字符，在这之中，**0** 就对应 **48**。那 **Unicode** 和 **ASCII** 有啥区别呢，抱着这个问题我了解到 **Unicode** 可以说是 **ASCII** 的扩展，**ASCII** 中的定义不变的基础上，**Unicode** 能支持定义更多的语言及各种符号比如汉语，阿拉伯语等等。

4) `Integer x = new Integer(18);`

```
Integer y = new Integer(18);
```

```
System.out.println(x == y);
```

```
Integer z = Integer.valueOf(18);
```

```
Integer k = Integer.valueOf(18);
```

```
System.out.println(z == k);
```

```
Integer m = Integer.valueOf(300);
```

```
Integer p = Integer.valueOf(300);
```

```
System.out.println(m == p);
```

首先明确一点引用类型的比较机制有两种，`==`比较的是对象，而 `equals()` 比较的是内容，此处三段代码皆使用的是 `==`

第 1 段代码的输出结果为 **false**，因为其都创建了一个新的对象，所以对象不同，为 **false**

第 2 段为 **true**，他们没有创立新的对象，而 **18** 是缓存在基本数据类型缓存池（**Integer**、**Byte**、**Short** 和 **Character** 等包装类在 **-128** 到 **127** 范围内的值缓存于其中）中的对象，故而其引用了同一对象，为 **true**

第 3 段为 **false**，基本数据类型缓存池范围是 **-128——127**，**300** 不属于该范围，所以是创立了两个不同的对象，为 **false**。

5)

对于 `int a = 5;`

```
int b = 7;
```

```
int c = (++a) + (b++);
```

```
System.out.println(c);
```

```
System.out.println(a + " " + b);
```

输出结果为

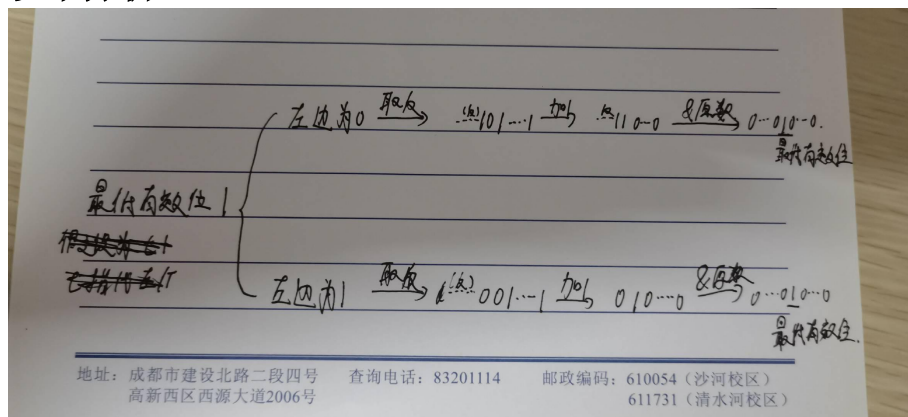
13

6 8

首先 a 前置自增先加 1 为 6 后参与运算, b 为后置自增先以 7 的值参加运算再加 1, 所以 c 为 13, a 为 6, b 为 8。其次 " " 代表了一个空格, 而 + 用于连接变量与符号。

6)

若 $a=0010$, 使用补码求 $-a$, 先求反, 得 1101, 再加 1, 得 $-a=1110$, $\&$ 表示每一对对应位进行与运算。只有当对应位都为 1 时, 结果位才为 1, 否则为 0。那么 $a \& (-a)$ 结果就为 0010。对于任意的非负整数 a , 式子 $a \& (-a)$ 表示的数是什么, 为什么得到这个结果, 我做了以下分析:



所以对于任意的非负整数 a , 式子 $a \& (-a)$ 表示的数为 a 的最低有效位。