1. 整型：byte,short,int,long；浮点型：float,double；字符型：char；布尔：boolean
2. byte一个字节：-128~127；short两个字节：-32768~32767；int四个字节：-2147483648~2147483647；long八个字节：-9223372036854775808~9223372036854775807
3. 自动类型转换；因为“0”的ASCII值为48，故int b=a+c相当于整数加法，也就是4+48=52，故输出为52。
4. 对于第一个输出结果为false。原因：通过new关键字创建的两个Integer对象不一样，在比较时，比较的是对象的引用不是对象的值，结果为false；

对于第二个输出结果为true。原因：Integer.valueOf()方法在-128 到 127会返回缓存的Integer对象，对于值为 18 的情况，在这个范围内，所以两次调用Integer.valueOf(18)返回的是同一个对象，比较时结果为true；

对于第三个输出结果为false。原因：当值超出 -128 到 127 的范围时，Integer.valueOf()方法每次都会创建一个新的Integer对象，所以m和p是不同的对象，比较结果为false。

1. ++a，是前置自增运算符，会先将 a 的值加 1，然后再参与运算。所以 a 的值变为 6； b++，是后置自增运算符，会先用现在的 b 参与运算，然后再将 b 的值加 1。这里先使用 b 的值 7 参与运算，执行完后，b 的值变为 8；最后计算 c 的值，即 c = 6 + 7 = 13。

System.out.println(c);输出 c 的值 13。

System.out.println(a+" "+b);输出 a 的值和 b 的值，但是中间有引号，引号里有空格，此时 a 的值为 6，b 的值为 8，所以输出结果为 “6 8”。6和8中间有一个空格。

1. a=0010，取反后为1101，加1得到1110，也就是-a，即-a=1110，然后将a和-a进行位与运算，得到0010。表示的数是a的二进制形式右边的1所代表的数值。
2. Float大约为±3.40282347E+38F，Double大约为±1.7976931348623157E+308。
3. 字符串是不能改变的对象。当在 append 方法中进行 s += " world" 操作时，实际上是创建了一个新的字符串对象，而原来传入的 s 所指向的字符串对象并没有被改变。在 main 方法中输出的 s 仍然是 "hello" 这个字符串对象，而在 append 方法中输出的是新创建的 "hello world" 这个字符串对象。