Day1：

**Windows system:**

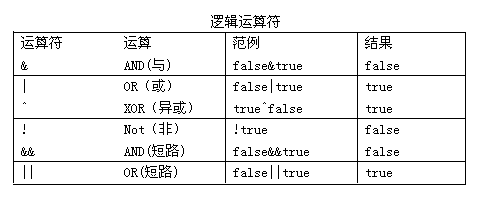
1. Win+up/down/left/right : adjust the size of window;
2. Alt+Tab : alter window;
3. Alt+shift : change language;

**Java**

1. 类中局部变量不用初始化；方法中的局部变量要初始化(by default);
2. Null 不可复制给基本类型(int,byte,short,long;float;string;boolen), 可以赋值给引用型(class;interface；数组)；
3. 判定用”==”判定是否为null；
4. Byte char short 参与运算 首先转换为 int 型；
5. Println中 在计算机检验到“之前，会优先考虑数学运算；
6. a++ 与++a；eg: if a=0, b = 1+(a++)+1=2;
7. Int a =10, a+=20, println(a)-->30
8. Int a = 1; short b =0; b+= a +1 ; 是正确的！！
9. Short b+= a + 1 (wrong)
10. char 类型用于运算 就是数值运算；
11. 只要byte short char 用于运算 都会强制转换成 int. (特殊情况除外);
12. Char b = ‘a’ + 1; (correct).
13. Int a =3;char b = a + 1; (wrong);

!!!14. int a = 10; char y = ‘a’ + 1; System.out.println(y);// (correct) result is : b

!!!15. char y = ’a’ + 1 ; System.out.println(y);//(correct) result is : b

1. Conclusion for 14 and 15 --> the character a in the ‘a’ is not the actually character a.
2. 

18.“&”和“&&”的区别：

单&时，左边无论真假，右边都进行运算；

双&时，如果左边为真，右边参与运算，如果左边为假，那么右边不参与运算。

“|”和“||”的区别同理，双或时，左边为真，右边不参与运算。

19.

|  |  |
| --- | --- |
| **位运算符的细节** | |
| << | 空位补0，被移除的高位丢弃。 |
| >> | 被移位的二进制最高位是0，右移后，空缺位补0；  最高位是1，最高位补1。 |
| >>> | 被移位二进制最高位无论是0或者是1，空缺位都用0补。 |
| & | 任何二进制位和0进行&运算，结果是0；  和1进行&运算结果是原值。 |
| | | 任何二进制位和0进行 | 运算，结果是原值；  和1进行 | 运算结果是1。 |
| ^ | 任何相同二进制位进行 ^ 运算，结果是0；  不相同二进制位 ^ 运算结果是1。 |

20.

.

|  |  |  |
| --- | --- | --- |
| **位运算符** | | |
| **运算符** | **运算** | **范例** |
| **<<** | **左移** | **3 << 2 = 12 --> 3\*2\*2=12** |
| **>>** | **右移** | **3 >> 1 = 1  --> 3/2=1** |
| **>>>** | **无符号右移** | **3 >>> 1 = 1 --> 3/2=1** |
| **&** | **与运算** | **6 & 3 = 2** |
| **|** | **或运算** | **6 | 3 = 7** |
| **^** | **异或运算** | **6 ^ 3 = 5** |
| **~** | **反码** | **~6 = -7** |

1. 三目运算

21(1). 条件表达式不能取代一般的if语句,仅当if语句中内嵌的语句为赋值语句(且两个分支都给同一变量赋值)时才能代替if语句.

21(2): 表达式1,表达式2,表达式3的类型可以不同.此时条件表达式的值的类型为它们中较高的类型.

main() {

　　　　　　　　　　char c1, ch;

　　　　　　　　　　ch = getchar();

　　　　　　　　　　c1 = ch <= 'Z' && ch >= 'A' ? ' @ ' : ch ;

　　　　　　　　　　putchar(c1);

　　　　　　　　}

22.

break 退出当前循环

continue 退出本次循环

23.