C++ 逻辑运算符 课后习题 - 答案

一、 填空题

1. `&&`，`||`，`!`
2. 全真才为真
3. 一真即为真
4. `false`，`true`
5. 短路求值 (或 短路)
6. `!` (非)，`&&` (与)，`||` (或)
7. 算 关 与

二、 单项选择题

1. **D**. `(5 > 3)` 为 `true`，`(2 < 1)` 为 `false`。`true && false` 的结果是 `false`，输出为 `0`。
2. **A**. `(10 == 9)` 为 `false`，`(3 != 4)` 为 `true`。`false || true` 的结果是 `true`，输出为 `1`。
3. **B**. `-1` 是非零数，被视为 `true`。`!true` 的结果是 `false`，即 `0`。
4. **C**. 题干描述了两个并列的条件，应该用 `||` 连接。“能被4整除但不能被100整除”是 `year % 4 == 0 && year % 100 != 0`，“能被400整除”是 `year % 400 == 0`。
5. **B**. `a` 是 `1` (true)，`b` 是 `0` (false)。`true || false` 的结果是 `true`，输出为 `1`。
6. **A**. 因为 `x > 1` (即 `5 > 1`) 的结果是 `true`，根据 `||` 的短路规则，后面的 `(y++ > 10)` 不会再执行，所以 `y` 的值保持 `10` 不变。
7. **A**. 因为 `x > 1` (即 `0 > 1`) 的结果是 `false`，根据 `&&` 的短路规则，后面的 `(y++ > 10)` 不会再执行，所以 `y` 的值保持 `10` 不变。
8. **D**. 一个数要么大于等于20，要么小于20，这两个条件覆盖了所有情况，所以用 `||` 连接起来结果永远为 `true`。
9. **B**. `!` 优先级最高，`!true` 变为 `false`。表达式变为 `false || false && true`。`&&` 优先级高于 `||`，先算 `false && true` 结果为 `false`。最后表达式变为 `false || false`，结果为 `false`，即 `0`。
10. **D**. `a > 1` (true) `&&` `a < 10` (false)，`true && false` 结果为 `false`。其他选项结果都为 `true`。

三、 判断题

1. √ (`&&` 的优先级高于 `||`，就像乘法高于加法一样。)
2. √ (当 `a` 为0时，`!a` 为 `true`，`a == 0` 也为 `true`。当 `a` 非0时，`!a` 为 `false`，`a == 0` 也为 `false`。)
3. × (它会先算 `5 < x`，结果是 `true`(1) 或 `false`(0)，然后再拿这个 `1` 或 `0` 去和 `10` 比较，永远为 `true`。正确的写法是 `x > 5 && x < 10`。)
4. × (关系运算符的优先级高于逻辑运算符。先判断大小关系，再进行逻辑组合。)
5. × (这是 `||` 的短路特性，只要 `A` 为 `true`，整个表达式就确定为 `true`，无需再计算 `B`。)
6. √ (`a > b` 即 `1 > 2` 为 `false`。`c > b` 即 `3 > 2` 为 `true`。`false || true` 结果为 `true`，输出 `1`。)

7. ×（`a` 是 `0`，即 `false`。对于 `&&` 运算，因为短路，`b++` 不会执行，`b` 的值仍然是 `10`。）
8. √（`0` 被视为 `false`，`!false` 的结果是 `true`，即 `1`。）
9. √（`true && false` 的结果是 `false`，`cout` 输出 `bool` 类型的 `false` 时会显示 `0`。）
10. √（这正是逻辑或 `||` 的定义。）