Introduction to Computer Science
Introductory Lectures

referring to chapter 1 of Gaddis, Starting Out With Java, quotes from later chapters and from Guzdial and Ericson, Intro. to Computing & Programming in Java, A Multimedia Approach
Computer programming is both an art and a science. It is an art because every program should be carefully designed. Here are a few of the things that must be considered for any real-world computer program:

- The logical flow of the instructions
- The mathematical procedures
- The layout of the programming statements
- The appearance of the screens
- The way information is presented to the user
- The program’s “user friendliness”
- Manuals, help systems, and/or other forms of written documentation
The || Operator

The || operator is known as the logical OR operator. It takes two boolean expressions as operands and creates a boolean expression that is true when either of the subexpressions is true. Here is an example of an if statement that uses the || operator:

```java
if (temperature < 20 || temperature > 100)
{
    System.out.println("The temperature is in the " + "danger zone.");
}
```

The message will be displayed if temperature is less than 20 OR temperature is greater than 100. If either relational test is true, the entire expression is true.

Designing the logical flow of the instructions

YOU, the programmer, design that the COMPUTER will COMPUTE the true/false answer to whether the temperature is below 20 OR is above 100

YOU also design that IF the COMPUTER'S answer is true

```java
{
    the COMPUTER shall print the text "The temperature is in the danger zone."
}
```

The COMPUTER prints nothing if not.
Designing the logical flow of the instructions
YOU, the programmer, design that the COMPUTER will COMPUTE the true/false answer to whether the temperature is below 20 OR is above 100.

YOU also design that IF the COMPUTER'S answer is true
{
   the COMPUTER shall print the text
   "The temperature is in the danger zone."
}
The COMPUTER prints nothing if not.
Designing the logical flow of the instructions

When determining whether a number is outside a range, it’s best to use the `||` operator. The following statement determines whether `x` is outside the range of 20 through 40:

```java
if (x < 20 || x > 40)
    System.out.println(x + " is outside the acceptable range.");
```

It’s important not to get the logic of these logical operators confused. For example, the boolean expression in the following if statement would never test true:

```java
if (x < 20 && x > 40)
    System.out.println(x + " is outside the acceptable range.");
```

Obviously, `x` cannot be less than 20 and at the same time be greater than 40.

```java
if (x < 20 || x > 40)
{
    System.out.println(x + " is out of ac. range.");
}
```

Gaddis had designed: (FIRST) the COMPUTER will COMPUTE the true/false answer to whether a number is below 20 OR is above 40 and then (SECOND), if that true/false answer is “true”, the COMPUTER will print a message including the value of that number.
Designing the logical flow of the instructions

It's important not to get the logic of these logical operators confused. For example, the boolean expression in the following if statement would never test true:

```java
if (x < 20 && x > 40)
    System.out.println(x + " is outside the acceptable range.");
```

Obviously, `x` cannot be less than 20 and at the same time be greater than 40.

“It's important not to get the logic ... confused. For example, the expression ... would NEVER test true:”

```java
if (x < 20 && x > 40)
    System.out.println(x + " is outside the acceptable range.");
```

Obviously, `x` cannot be less than 20 and at the same time be greater than 40.”
“It's important not to get the logic ... confused. For example, the expression ... would NEVER test true:”

```java
if (x < 20 && x > 40) {
    System.out.println(x + " is out of ac. rnge.");
}
```

Obviously, x cannot be less than 20 and at the same time be greater than 40.”

IClicker question 1: Will the computer print some number is out of ac. rnge. ?

(A) Always
(B) Sometimes but not always
(C) Never
“It's important not to get the logic ... confused. For example, the expression ... would NEVER test true:"

```java
if (x < 20 && x > 40) {
    System.out.println(x + "is out of ac. rnge.");
}
```

Obviously, \(x\) cannot be less than 20 and at the same time be greater than 40.”

IClicker question 2: The computer **never** prints some number **is out of ac. rnge.**

WHY NOT?
(A) \(x\) cannot be less than 20
(B) \(x\) cannot be greater than 40
(C) \(x<20\) together with \(x>40\) **is mathematically impossible.**
(D) \(x\) is always between 20 and 40 inclusively.
“It's important not to get the logic ... confused. For example, the expression ... would NEVER test true:"
if (x < 20 && x > 40)
{
    System.out.println(x + " is out of ac. rnge.");
}
Obviously, x cannot be less than 20 and at the same time be greater than 40.”
The computer never prints ... out of ac. rnge.

BECAUSE
(C) x<20 together with x>40 is mathematically impossible.
“It's Obviously, $x$ cannot be less than 20 and at the same time be greater than 40.”

The computer never prints ... out of ac. range.  

BECAUSE

(C) $x<20$ together with $x>40$ is mathematically impossible.

The value of $x$ is could be ANY NUMBER WHATSOEVER.

It might be $<20$, be equal to 20, $>20$ AND $<40$ (like 31), $\equiv 40$ (be equal to 40), or $>40$.

But, because numbers are conceptualized on the famous linearly-ordered number lines, a single number $x$, even one you don't know, cannot have BOTH $x<20$ AND $x>40$. 
After carefully reading this quotation, and only if necessary, other content of sec. 6.5 (Logical Operators) of the UA book (sec. 3.5 in Gaddis' full book), answer this question:

Suppose the programmer made the mistake of writing the symbol `||` instead of `&&` in the (above) statement repeated here:

```java
if (x >= 20 && x <= 40)
    System.out.println(x + " is in the acceptable range.");
```

That's a typical mistake both beginners and professionals make.

Answer BOTH (A) What will the computer do (or not do)? (B) Explain the logic behind your figuring out the answer to (A).
Designing the logical flow of the instructions

Logical flow means two things:
• More complex conditions computed from simpler ones (we saw examples)

• The ORDER in which the robot, Turtle or other agent does things, which is generally the ORDER in which the programmer had pre-written them