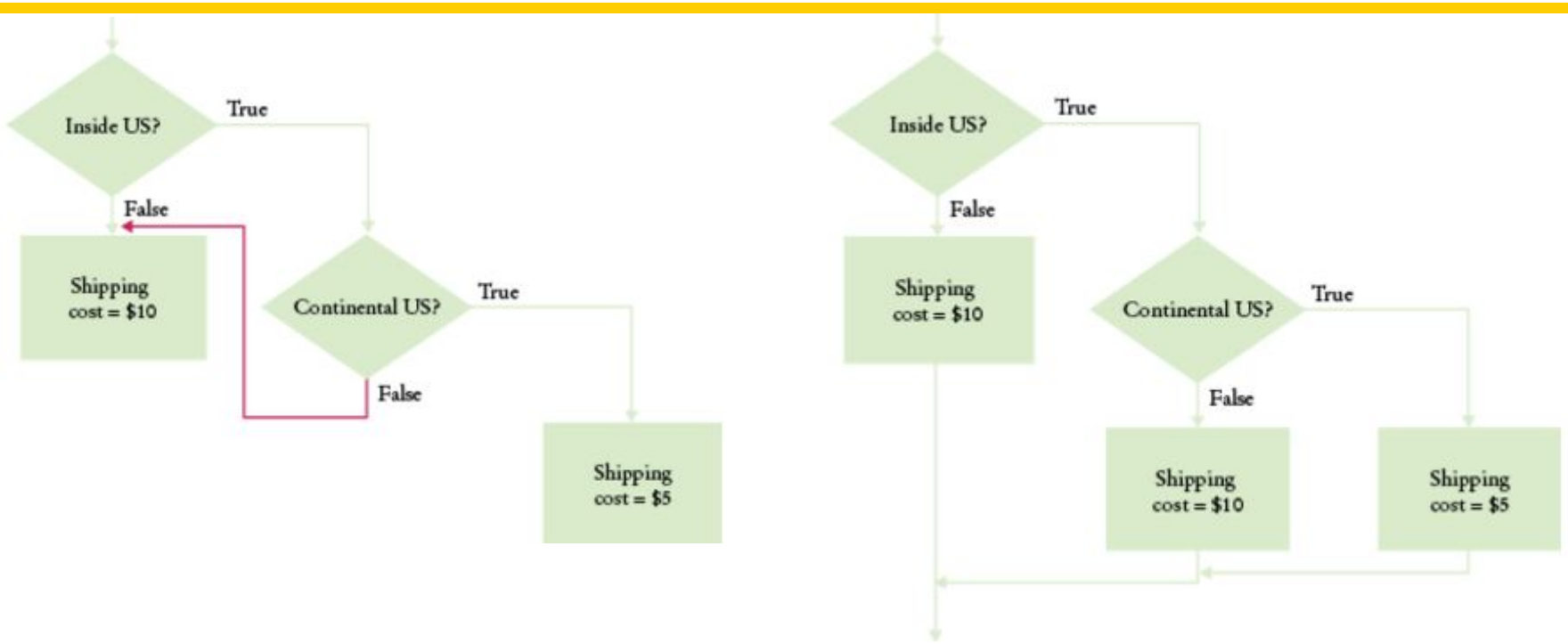
Topic 5

1. The `if` statement
2. Comparing numbers and strings
3. Multiple alternatives
4. Nested branches
5. Problem solving: flowcharts
6. Problem solving: test cases
7. Boolean variables and operators
8. Application: input validation
9. Chapter summary

Creating Flowcharts

- Drawing a flowchart can help you understand and code a decision problem
  - Though they can get too large in some cases, so use pseudocode if the flowchart grows too long.
- Use the 3 standard flowchart boxes:
  - Rectangle = simple task
  - Parallelogram = input/output
  - Diamond = decision (or condition)
    - Label the lines out of the diamond with the true/false value
- To avoid spaghetti code (overly complex branching)
  - *Never point an arrow inside another branch*

Don't Draw an Arrow to Another Branch Arrow



- The version on the left shares the "shipping cost=$10" task box, but leads to complex code, which is not easily modified if say, the cost for one of the decisions changed.
- Add the extra box and avoid the double branch (red arrow), as shown on the right.

Topic 6

1. The `if` statement
2. Comparing numbers and strings
3. Multiple alternatives
4. Nested branches
5. Problem solving: flowcharts
6. Problem solving: test cases
7. Boolean variables and operators
8. Application: input validation
9. Chapter summary

Prepare Test Cases Ahead of Time

- To test your program, you cannot try out all possible inputs
- If the program correctly computes one or two tax amounts in a given bracket, then we can assume it is correct.
- You should also test on the *boundary conditions*, at the endpoints of each bracket

this tests the < vs. <= situations.

Test Cases, Boundary Values

- Here are some possible test cases for the tax program:

Test Case	Expected Output	Comment
30,000 s	3,000	10% bracket
72,000 s	13,200	3,200 + 25% of 40,000
50,000 m	5,000	10% bracket
10,400 m	16,400	6,400 + 25% of 40,000
32,000 m	3,200	boundary case
0 s	0	boundary case

The Dangling `else`

- When an `if` statement is nested inside another `if` statement, the following error may occur.
- Can you find the problem with the following?

```
double shipping_charge = 5.00;
                  // $5 inside continental U.S.
if (country == "USA")
   if (state == "HI")
      shipping_charge = 10.00;
                  // Hawaii is more expensive
else // Pitfall!
   shipping_charge = 20.00;
                  // As are foreign shipments
```

The Dangling `else` Problem

- The indentation level *seems* to suggest that the **`else`** is grouped with the test **`country == "USA"`**.
- Unfortunately, that is not the case.
- The compiler *ignores* all indentation and matches the **`else`** with the preceding **`if`**.

```
double shipping_charge = 5.00;
                  // $5 inside continental U.S.
if (country == "USA")
   if (state == "HI")
      shipping_charge = 10.00;
                  // Hawaii is more expensive
else // Pitfall!
   shipping_charge = 20.00;
                  // As are foreign shipments
```

The Dangling `else` Problem – The Solution

- So, is there a solution to the dangling **else** problem.
- You can put one statement in a block, inside braces.

```
double shipping_charge = 5.00;
                  // $5 inside continental U.S.
if (country == "USA")
{
   if (state == "HI")
      shipping_charge = 10.00;
                  // Hawaii is more expensive
}
else
   shipping_charge = 20.00;
                  // As are foreign shipments
```