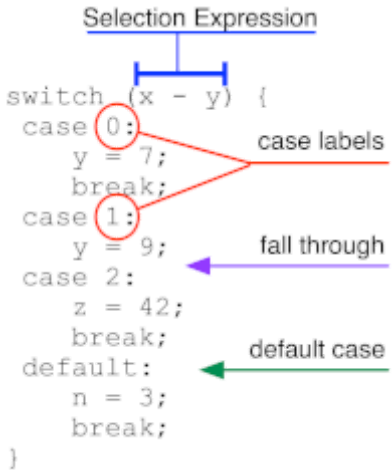
Switch/Case

coursera.org/learn/programming-fundamentals/supplement/pu8s1/switch-case

switch/case

Another way that programs can make decisions is to use switch/case. The syntax of switch/case is shown in the figure below. Here, when the execution arrow reaches the switch statement, the selection expression—in parenthesis after the keyword switch—is evaluated to a value. This value is then used to determine which case to enter. The execution arrow then jumps to the corresponding case—the one whose label (the constant immediately after the keyword case) matches the selection expression's value. If no label matches, then the execution arrow jumps to the default case if there is one, and to the closing curly brace of the switch if not.

Once the execution arrow has jumped into a particular case, execution continues as normal until it encounters the keyword break. When the execution arrow reaches the break keyword, it jumps to the close curly brace which ends the switch statement. Note that reaching another case label does not end the current case. Unless the execution arrow encounters break, execution continues from one statement to the next. When the execution arrow passes from one case into the next like this, it is called "falling through" into the next case.



For example, if we were executing the code in the figure above, and reached the switch statement with x having a value of 17 and y having a value of 16, then we would first evaluate the selection expression (x - y), and get a value of 1. The execution arrow would then jump to case 1: and begin executing statements after it. We would execute y = 9;. Then we would fall through the next case label—our execution arrow would move past it into the next case (the label itself has no effect). Then we would execute z = 42;. Next, we would execute the break; statement, causing our execution arrow to jump to the close curly brace of the switch, after which we would continue executing whatever other statements are there.