AP Computer Science A**:** Boolean Expressions and if Statements**:** 03.04: Condition Statements: if-else

**Decisions, Decisions**

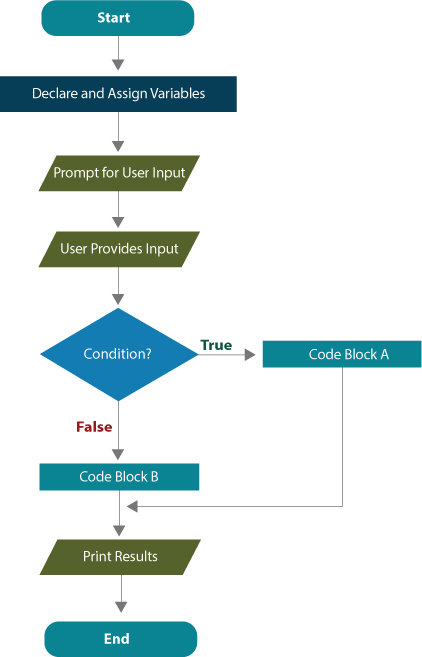
Consider the difference between the following two sentences.

If the hours are greater than 40,  
then the rate of pay is time-and-a-half.

If the hours are greater than 40,  
then the rate of pay is time-and-a-half,  
otherwise the rate of pay is normal.

The second sentence explicitly states what the rate of pay will be when someone works more than 40 hours and when they work 40 hours or less. The first sentence indicates overtime will be paid for more than 40 hours of work, but only implies that normal pay applies for working 40 hours or less. This subtle distinction would probably not make much difference in a conversation between two people, but such differences can be the source of great confusion in a computer program.

### Part 1



Frequently you will want to execute a specific [block of code](javascript:void(0);) if a condition is true, but a different block if the condition is false. A simple **if** statement won’t always work well in that situation.

When there are two possible alternatives based on whether a condition is true or false, the **if-else** version of the condition statement is used as illustrated in the accompanying flowchart.

* If the condition is true, execution branches to Code Block A.
* If the condition is false, execution branches to Code Block B.

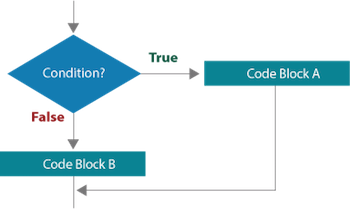
Blocks of code consist of one or more Java statements.

The Java compiler will notify you immediately if a condition statement contains a syntax error; however, you will not be so lucky when logic errors exist. Logic errors will either not be caught at all, or show up somewhere else in the program. Consequently, logic   
errors are often difficult to track down and may take time-consuming detective work.

### Part 2

The structure of a generic **if-else** statement is shown below, along with a corresponding flowchart segment.

…



**if(boolean Condition)**  
**{**  
**execute Code Block A;**  
**}**  
**else**  
**{**  
**execute Code Block B**  
**}**

…

Syntactically, it is important to note that the condition must be **boolean**; it must be written inside of a pair of parentheses, and each line in an executable block must end with a semicolon. Notice, however, that the actual **if** and **else** statements **do not** end with a semicolon. The start and stop of each block of code is indicated with a pair of curly braces. In terms of style, it is good programming practice to indent statements within each block of code.

### Part 1

Learning to write computer programs involves constantly assessing alternative coding strategies as you learn new techniques. More advanced techniques build upon understanding simpler ways of doing things, and this is the case with learning to use the three variations of the condition statement.

### Part 2

There are subtle differences between **if** and **if-else** statements that are critical to note from the practical standpoint of writing code, as well as the logical design of algorithms.

### Part 3

The SalaryV1 class can also be modified to use the more versatile **if-else** statement. Study the modifications carefully so you can apply these new techniques in your programs.

### Part 4

Since the examples of conditional statements have only dealt with numeric values, you may be wondering whether **if** statements can handle alphanumeric data. The answer is a little bit fuzzy: yes and no! Relational operators (<, >, <=, >=, ==, !=) have a specific function in Java; they can be used with primitive data types to compare their values. Strings are objects; therefore the relational operators won't work for evaluating their values. Since chars are primitive, we can work with those for now.