Practice Exercises for Logic and Conditionals

Solve each of the practice exercises below. Each problem includes three CodeSkulptor links: one for a template that you should use as a starting point for your solution, one to our solution to the exercise, and one to a tool that automatically checks your solution.

1. Write a Python function `is_even` that takes as input the parameter `number` (an integer) and returns `True` if `number` is even and `False` if `number` is odd. Hint: Apply the remainder operator to `n` (i.e., `number % 2`) and compare to zero. Even template --- Even solution --- Even (Checker)
2. Write a Python function `is_cool` that takes as input the string `name` and returns `True` if `name` is either `"Joe"`,`"John"` or `"Stephen"` and returns `False` otherwise. (Let's see if Scott manages to catch this. ☺) Cool template --- Cool solution --- Cool (Checker)
3. Write a Python function `is_lunchtime` that takes as input the parameters `hour` (an integer in the range [1,12]) and `is_am` (a Boolean "flag" that represents whether the hour is before noon). The function should return `True` when the input corresponds to 11am or 12pm (noon) and `False` otherwise. If the problem specification is unclear, look at the test cases in the provided template. Our solution does not use conditional statements. Lunchtime template --- Lunchtime solution --- Lunchtime (Checker)
4. Write a Python function `is_leap_year` that take as input the parameter `year` and returns `True` if `year` (an integer) is a leap year according to the Gregorian calendar and `False` otherwise. The Wikipedia entry for leap yearscontains a simple algorithmic rule for determining whether a year is a leap year. Your main task will be to translate this rule into Python. Leap year template --- Leap year solution --- Leap year (Checker)
5. Write a Python function `interval_intersect` that takes parameters `a`, `b`, `c`, and `d` and returns `True` if the intervals [a,b] and [c,d] intersect and `False` otherwise. While this test may seem tricky, the solution is actually very simple and consists of one line of Python code. (You may assume that $a \leq b$ and $c \leq d$.) Interval intersect template --- Interval intersect solution --- Interval intersect (Checker)
6. Write a Python function `name_and_age` that take as input the parameters `name` (a string) and `age` (a number) and returns a string of the form `"% is % years old."` where the percents are the string forms of `name` and `age`. The function should include an error check for the case

when `age` is less than zero. In this case, the function should return the string `"Error: Invalid age"`. Name and age template --- Name and age solution --- Name and age (Checker)

7. Write a Python function `print_digits` that takes an integer `number` in the range [0,100) and prints the message `"The tens digit is %, and the ones digit is %."` where the percents should be replaced with the appropriate values. The function should include an error check for the case when `number` is negative or greater than or equal to 100. In those cases, the function should instead print `"Error: Input is not a two-digit number."`. Print digits template --- Print digits solution --- Print digits (Checker)
8. Write a Python function `name_lookup` that takes a string `first_name` that corresponds to one of (`"Joe"`, `"Scott"`, `"John"` or `"Stephen"`) and then returns their corresponding last name (`"Warren"`, `"Rixner"`, `"Greiner"` or `"Wong"`). If `first_name` doesn't match any of those strings, return the string `"Error: Not an instructor"`. Name lookup template --- Name lookup solution --- Name lookup (Checker)
9. Pig Latin is a language game that involves altering words via a simple set of rules. Write a Python function `pig_latin` that takes a string `word` and applies the following rules to generate a new word in Pig Latin. If the first letter in `word` is a consonant, append the consonant plus `"ay"` to the end of the remainder of the word. For example, `pig_latin("pig")` would return `"igpay"`. If the first letter in `word` is a vowel, append `"way"` to the end of the word. For example, `pig_latin("owl")` returns `"owlway"`. You can assume that `word` is in lower case. The provided template includes code to extract the first letter and the rest of `word` in Python. Note that, in full Pig Latin, the leading consonant cluster is moved to the end of the word. However, we don't know enough Python to implement full Pig Latin just yet. Pig Latin template --- Pig Latin solution --- Pig Latin (Checker)
10. **Challenge:** Given numbers a, b, and c, the quadratic equation $ax^2+bx+c=0$ can have zero, one or two real solutions (i.e; values for x that satisfy the equation). The quadratic formula $x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$ can be used to compute these solutions. The expression b^2-4ac is the *discriminant* associated with the equation. If the discriminant is positive, the equation has two solutions. If the discriminant is zero, the equation has one solution. Finally, if the discriminant is negative, the equation has no solutions. Write a Python function `smaller_root` that takes an input the numbers `a`, `b` and `c` and returns the smaller solution to this equation if one exists. If the

equation has no real solution, print the message `"Error: No real solutions"` and simply return. Note that, in this case, the function will actually return the special Python value `None`.

Quadratic root template --- Quadratic root solution --- Quadratic root (Checker)