

Text: [How to think like a computer scientist \(learning with Python¹\)](#).

Reading Assignment 4: [Chapter 4 \(Conditionals and recursion\)](#).

Questions: Evaluate $(a / b) * b + a \% b$ where a and b are integer variables². How can you determine if an integer number is odd/even? Write a program that determines the coins that a cashier will need to return an amount of money originally expressed in cents (e.g., a change of 123 cents will be returned as 4 quarters, 2 dimes, 0 nickels and 3 cents). What is $3 / 4$? What is $3 \% 4$? How do you calculate the last digit of any given number?

What is a `boolean` expression? Is $2 + 3$ a boolean expression? What value does it have? Is $2 > 3$ a boolean expression? What value does it have? If x is 3 and $x \leq y$ is `True` what values can y have? How do you read the expression $a \neq b$? What does it mean? Does it have a value? If so, what is it? Is $x = y$ a boolean expression³?

Explain the following results:

```
>>> x = 1
>>> y = 2
>>> x and y
2
>>> x or y
1
>>>
```

Simplify the following expressions:

- `a and not a`
- `a or not a`
- `x >= 6 and x < 7`
- `a == False`
- `a == True`
- `a or True`
- `a and True`
- `a and False`
- `a or False`
- `False and False or True`
- `False and (False or True)`

Is `not (a and b)` the same as `not a or not b`? Justify your answer.

¹ <http://www.cs.indiana.edu/classes/a202-dger/fall2006/a201-text/thinkCSpy.pdf>

² Do you *need* to know the values of a and b ? Why or why not?

³ If not, what kind of expression is it?

What are conditional statements and what can they do for us? Can you find the largest of two numbers without using⁴ an `if` statement? What if the two numbers are integers?

Describe the concepts of `block` and `body`. What is indentation and how is it related to block and body? Explain the use of the `pass` statement. Illustrate it with an example. What do we mean by alternative⁵ execution?

Consider the following function:

```
def what(n):
    if n == 0:
        return 0
    else:
        return n + what(n-1)
```

What does it do? What's the result returned by `what(3)`? What does `what(-3)` evaluate to? How can you change `what` so it works for negative arguments as well?

Define `chained conditionals` and write a program using a chained conditional to solve the following problem: Write a program (called `Eight`) that translates a number between 0 and 4 into the closest⁶ letter grade. For example, the number 2.8 (which might have been the average of several grades) would be converted to B-. Break ties in favor of the better grade; for example 2.85 should be a B.

What is a nested conditional?

Consider the following code:

```
if x > 3:
    if x <= 5:
        y = 1
    elif x != 6:
        y = 2
    else:
        y = 3
else:
    y = 4
```

If `y` has the value 2 after executing the above program fragment, then what do you know about the initial value of `x`?

⁴ You can use `+`, `-`, `/`, and `abs(...)` that is: addition, subtraction, division and absolute value.

⁵ Explain what we mean by “branches in the flow of execution”.

⁶ A, B, C, D have the values 4, 3, 2, 1. A plus adds 0.3, a minus takes 0.3 away. F is 0. A+ is a 4.

Now consider the following two fragments:

```
if x == 5:
    x = x + 1
else:
    x = 8
```

```
if x == 5:
    x = x + 1
if x != 5:
    x = 8
```

Are the two fragments logically equivalent? Why or why not?

Simplify `a <= b and b <= c` in Python.

Consider the following Python code:

```
def what(n):
    if n == 0:
        result = 0
    else:
        print n
        result = n + what(n-1)
    return result
```

What's the result of calling `what(10)`? Now swap the statements marked [1] and [2]:

```
def what(n):
    if n == 0:
        result = 0
    else:
        result = n + what(n-1)
        print n
    return result
```

What's the result of calling `what(10)` now?

What is recursion? Write a recursive function that plays “guess the number” with the user. Use `raw_input` for user input and keep playing until the user guesses the number. Can you keep track of and report the total number of guesses at the end? Can you program the game where you have a total of ten tries at most? Here's a start:

```
def game(____):
    ____ = raw_input("Enter guess: ")
    if ____ == ____:
        print "You guessed it. "
    else:
        if ____ < ____:
            print "Try higher"
        else:
            print "Try lower. "
    ____(____)
```

Did you use stack diagrams to answer the two questions about `what` on this page?

Define a function called `sum` with no arguments that asks the user (using `raw_input`) for two integers and prints their sum.

Reading Assignment 5: Chapter 5 (Fruitful functions).

Questions: What does `abs` do? Write a function called `distance` which receives four arguments (`x0`, `y0`, `x1`, `y1`) representing the coordinates of two points in the plane and calculates and returns the distance⁷ between them. Define and use the function `compare` described on page 48. Write a function that calculates the area of a triangle using Heron's formula⁸:

In **geometry**, **Heron's formula** (also called **Hero's formula**) states that the **area** of a **triangle** whose sides have lengths a , b and c is

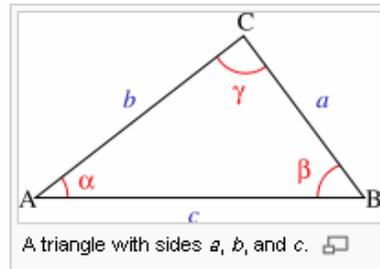
$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

where s is the triangle's **semiperimeter**:

$$s = \frac{a + b + c}{2}$$

(see also **square root**). Heron's formula can also be written

$$\text{area} = \frac{\sqrt{(a+b+c)(a+b-c)(b+c-a)(c+a-b)}}{4}.$$



What is composition? Use composition to implement a function that calculates the area of a triangle using Heron's formula but starting from the coordinates of the three vertices.

Write a function that receives three arguments representing the lengths of its sides and determines if the triangle is isosceles⁹ or not. Write a function called `inside` that receives information about a circle (two coordinates for the center and the value of the radius) and a point (two coordinates) and returns a boolean value indicating if the point is inside the circle or not.

Write a function that determines if two circles¹⁰ overlap. In this last case you have the freedom and responsibility of deciding what (and how many) arguments the function has.

Reading Assignment 6: Chapter 6 (Iteration).

Questions: If `i` is an integer variable, how do you increment¹¹ it?

⁷ Which is the square root of the sum of the squared differences between the corresponding coordinates.

⁸ http://en.wikipedia.org/wiki/Heron's_formula

⁹ Same question for equilateral, right, scalene.

¹⁰ Same question about rectangles.

¹¹ What does *that* mean?

Take a look and review the material on page 56 then find/devise a non-recursive way to produce the first few Fibonacci numbers.

What is the output of the following code fragment:

```
n = 10
m = n
n = 12
print m
```

Does the change in `n` affect `m` at all? Why (or why not)?
Briefly explain¹² and evaluate (assume `i` holds an integer value):

- `i = i + 1`
- `i == i + 1`
- `i == i`
- `i = i`

What does the following code fragment print¹³?

```
print 1,
print 2,3
print 4,5,6
print 7,
print "ha!"
```

What does it mean for equality to be commutative? How is assignment different from equality¹⁴? Why is assignment non-commutative?

What is iteration? Is iteration the same as repetition? Python provides a `while` statement; what is its purpose? Write a function that takes one argument (call it `n`) and prints the integers between 1 and `n` in ascending order; write the function recursively, then using a `while` loop. Repeat this exercise but make the function print the numbers in descending order this time.

Consider the following Python code:

```
i = 10
while i < 10:
    i = i - 1
print i
```

What does it print (or amount to)? Do you see any potential¹⁵ problem with it?

¹² Discuss differences, similarities of purpose and evaluation/processing.

¹³ Try to predict with great accuracy the output, spaces and everything.

¹⁴ Isn't `a = b` essentially an assignment but *also* a statement of equality?

¹⁵ What if `i` starts at 9 instead of 10?

Now consider the following code:

```
s = 0
i = 10
while i > 0:
    s = s + i
    i = i - 1
print s
```

What does it print (if anything)? What do you think it is computing?

How do you find if a number divides another? What is an odd number? Write a function that takes one argument (a positive integer) and reports if the argument is odd or not. What is a prime number? Write a function that takes one argument (a positive integer) and reports if the argument is prime or not.

Can you calculate the sum of the first n numbers without using a loop or recursion?

Is the loop in the code below infinite? How do you know (for sure) before you run it?

```
m = 3
n = 5
while n < 10:
    m = n - 1
    n = 2 * n - m
print n, m
```

Write a program that prints a table on two columns: on the left the integer temperatures between 0 and 100 (Fahrenheit) and in the right column the corresponding Celsius values. Make a similar table that helps converting miles into kilometers and pounds in kilograms.

How do you print a `tab` character¹⁶? What is an escape character¹⁷? How do you print a newline? How do you print three new lines using only one print statement? What do we mean by `cursor`? What base is the log function in Python using? How can you calculate the logarithm in base 2 of a number?

Write a one line program in Python that produces this output:

```
+---+
|   |
+---+
```

Solve the exercise at the end of section 6.3.

¹⁶ What are tabs useful for?

¹⁷ Can you give an example?

Write functions to print scalable letters **Z**, **M**, **E**, **L**, **C**, **F**, **W** and the digit **4** as square patterns¹⁸ of size `n` (where `n` is the only argument of the function, the size).

Here's an eight by eight **Z**: remember the function only gets the size of the pattern:

```

* * * * * * * *
      *
     *
    *
   *
  *
 *
* * * * * * * *

```

Also write programs that generate each of these three patterns as scalable patterns:

```

* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *
* * * * * * * * * *      * * * * *      * * * * * * * * * *

```

What do we mean by a loop variable? Give examples. How many loop variables are you using in your scalable patterns programs? Are you using encapsulation in your scalable patterns programs? Is it easier or harder to use encapsulation (also, is it better or worse)? Is it always easier/better/worse/harder to use encapsulation? Give an example of each.

Write a program that asks the user for two integer numbers and stores them in variables `n` and `m` and with `n <= m` and prints all the prime numbers between `n` and `m`. Are you using encapsulation and generalization in your coding¹⁹?

Write a program that prints a square table of random numbers (where the size provided).

What does the term `development plan` mean on page 66 in the text?

What is a local variable? Is a local variable like a parameter? In what ways is a local variable not like a parameter²⁰? Why do we need local variables at all in Python? What local variables have you seen in the code written so far in this material?

Write stack diagrams to describe the action of the following code:

```
def odd(num):
```

¹⁸ See <http://www.cs.indiana.edu/classes/a201-dger/spr2003/notes/labSix.html>
¹⁹ The circumstances are very suitable, in this case.
²⁰ Can you use a variable before it's initialized? How do parameters get initialized?

```

    if num == 0:
        return False
    else:
        below = num - 1
        return even(below)

def even(num):
    if num == 0:
        return True
    else:
        below = num - 1
        return odd(below)

print odd(4)
print even(4)
print odd(5)
print even(5)

```

How could you fix the code so it works for negative inputs as well? Is `below` an example of a local variable in the sense discussed on page 67 in the text (same name, different functions)? Is there any interference? Why or why not? Is `num` a similar example? Is it also a local variable? What is it? What's the difference, if any?

Write a program that calculates for every number between `n` and `m` (given) the number of times it takes the `sequence` function on page 61 to reach 1 (one). Change the program so it also prints a histogram, like this:

```

>>> histogram(5, 14)
14 ***** (17)
13 ***** (9)
12 ***** (9)
11 ***** (14)
10 ***** (6)
 9 ***** (19)
 8 *** (3)
 7 ***** (16)
 6 ***** (8)
>>>

```

Solve the exercise on page 69 and explain what you did and how.

Summarize the benefits of functions and give an example for benefit²¹.

Turn in a Word document with all the questions answered completely.

²¹ The book also gives a list of benefits. Feel free to discuss them if you decide to rely on their list.