

EXAMPLE: drawing the stack

Consider the following code:

```
1  def main():
2      string = "we love comp sci!"
3      letter = "e"
4      result = count(letter, string)
5      print("Count of %s = %d" % (letter,result))
6
7  def count(ch, s):
8      total = 0
9      for i in range(len(s)):
10         if s[i] == ch:
11             total = total + 1
12         # draw stack to this point!
13         # show the stack as it would look,
14         # just before the return statement
15
16     return total
17
18  main()
```

Follow these steps to draw the stack:

- draw the empty stack
- put the first function called on the stack (usually **main()**)
 - * allocate parameters (if any) for this stack frame (assign to arguments)
 - * set up any locally declared variables (you can usually skip drawing any loop variables, since they update frequently and often clutter the diagram)
 - * execute the function, step by step
 - * send the return value back to the calling function
 - * remove the finished function from the stack (erase or cross it out)
- continue executing the function that is *now* on the top of the stack

Below are the steps for the above code example, with line numbers, so you can see what each line is doing. Try drawing the stack and see if you get the same drawing!

```

Line #   do this
-----
18       put a frame for main() on the stack
2        put variable string in the frame, draw arrow
         to the value "we love comp sci!!"
3        put letter in the frame, draw arrow to "e"
4        count() function is called, so put frame
         for count() on top of the stack
7        put parameter/variable ch in count() frame,
         draw arrow to value "e"
7        put parameter/variable s in count() frame, draw
         arrow to value "we love comp sci!!"
8        put local variable total in count() frame, draw
         arrow to value 0
9        for loop starts, but we will ignore the variable i
11       update total to point to 1 (first time s[i]==ch)
11       update total to point to 2 (second time)

12       Comment says to stop here.
         Your stack should look like this:

```

