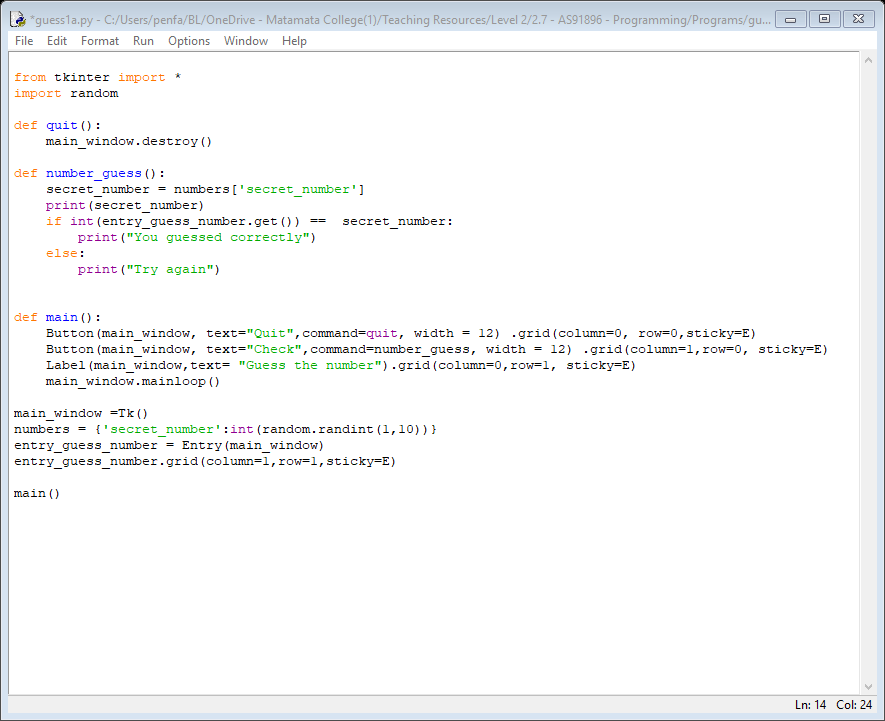
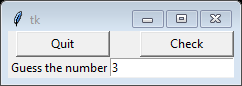
# making a guess the number game

* This is a bit of a classic game when learning programming
* You get the computer to create a random number then you try and guess it
* Make sure you don’t overwrite your previous program as we will be coming back to it

 *Python 3.7.0 Shell*

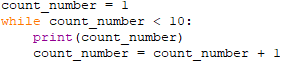
Task time

* Get it to print “That guess is too high” or “That guess is too low”
* Can you make any other improvements to the game



# using the while statement

* One of the most common features of programs is the ability to loop
* We are going to start without the GUI then get back to it soon
* Again, to save space, comments are not included in the samples. You should always include them

Python 3.7.0 Shell

Task time

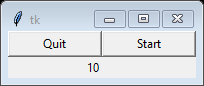
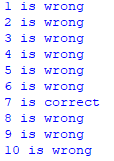
* Change the program so it counts from 1 to 10, not 1 to 9
* Reverse the count so it goes from 10 to 1

## While operators

* Just like the if statement you can use the operators == < > <= >= !=

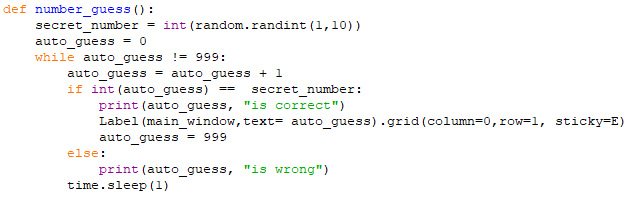
## Automating “guess the number”

* Admittedly this defeats the whole purpose of the game, but we will do it anyway
* Instead of trying all the possible numbers we will get it to automatically start at 1
* Hopefully this is not how you have been doing it!

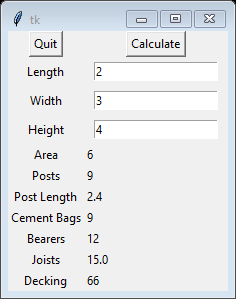
Task time

* Tidy up some issues with the program
* Improve the program so it stops when it gets to the correct number
* Add in a delay so you can see it scroll through the numbers (search “python time.sleep”)



* Note the change to **!= 999** this provides a way to exit from the loop
* Try removing the sleep and set the random to 1,000, 000 and see how long it takes
* Is it quicker if you also remove the print?
* Is it even quicker if you run in native Python, not Idle?

Major task – deck planner

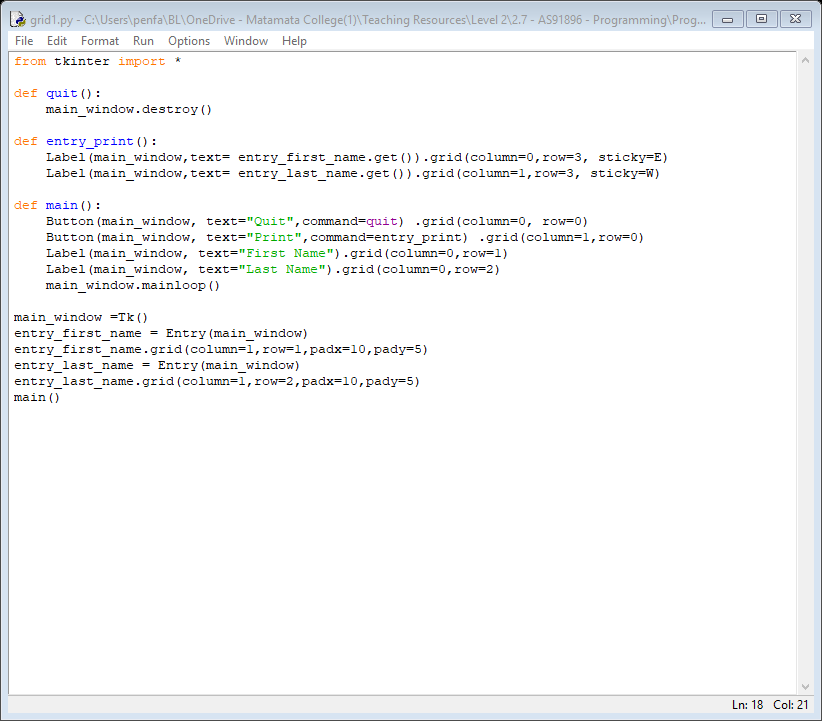
* It is now time to do a program of your own
* If you know how to do it just look at the specifications and write the program
* Otherwise, follow the step by step tasks to get it done
* For this task your user puts in the width, length and height of the deck and the program will tell you all the materials you need
* This is an illustration only - seek professional help if really building a deck

specifications

* The area is calculated length \* width
* Posts required is (length +1 ) \* width
* The length of the posts is 1.8 for height 1 or less, 2.4 if greater
* Bags of cement is the same as the number of posts
* Bearers are area \* 2
* Joists are area /.4
* Decking is area \* 11

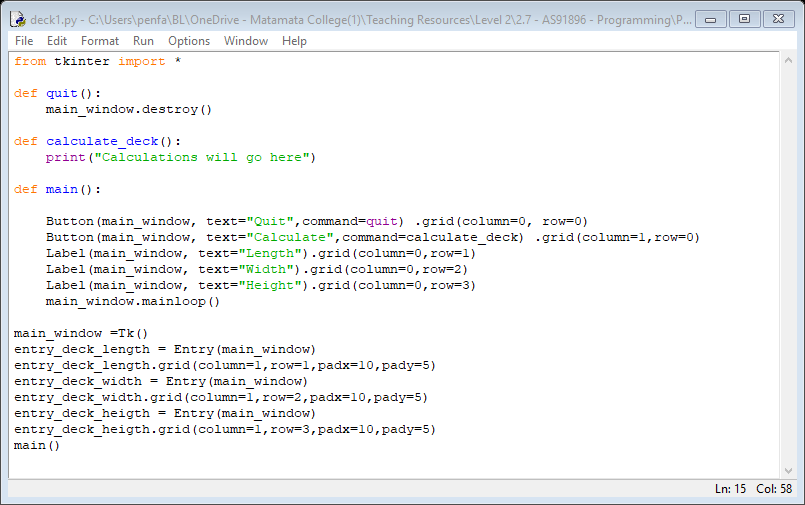
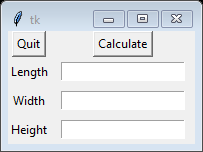
Finding a starting point

* Hopefully you have saved a basic GUI program that you can start with - this is a good one



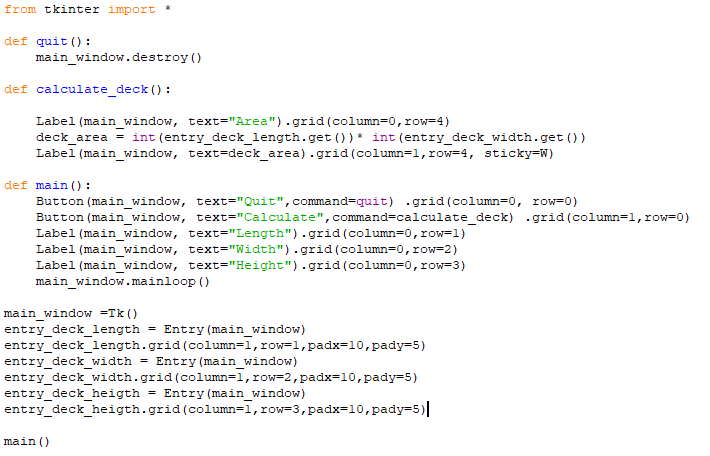
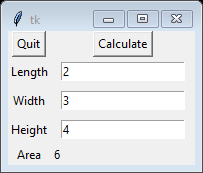
entry and variable names

* Change the program first so all the variable names are correct, and you have the inputs
* Only look at the program below if you get stuck – be a programmer, not a typist

Start the calculations

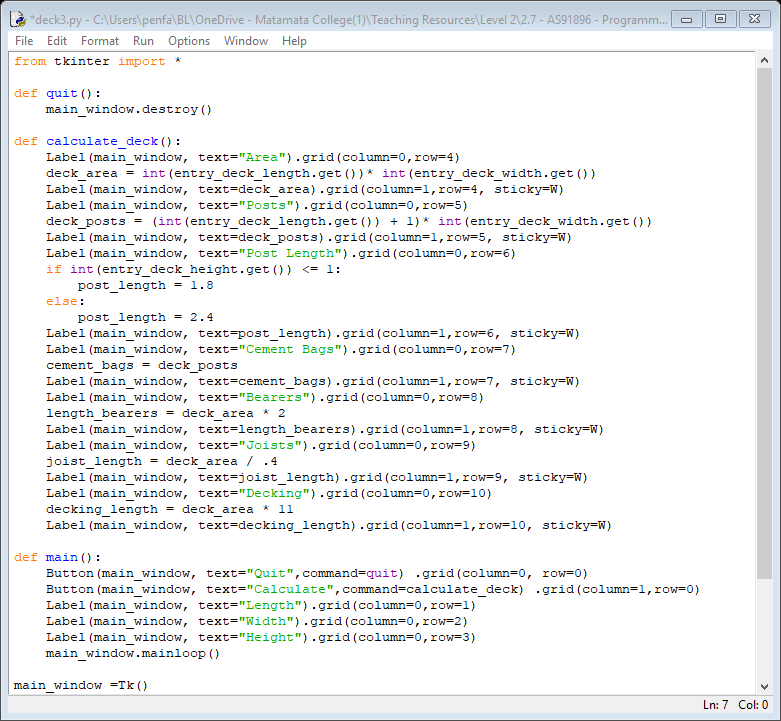
* Start by just doing the first one or two calculations to make sure it looks right

* See if you can get program finished now - on the next page is a poor solution

## poor solution

* This program works, but only shows the row headings after you click on Calculate
* Also, there are no comments or constants – you need to include these

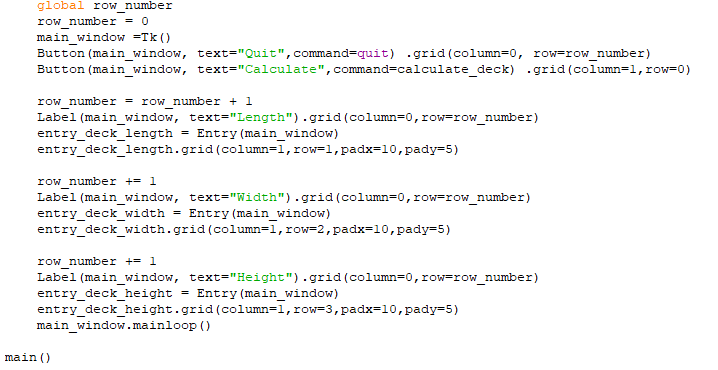


task time

* Improve the program so all the labels for the items are shown before you click on Calculate
* Also, if you have not commented your program, do it now

# tidying up grids

* When you did the deck planner you may have found it difficult keeping up with the row number
* It will cause even more issues when we get onto lists, so we need a better way
* Instead of putting in the numbers, it is better to use a variable that we keep increasing
* Column numbers are not the same issue with the programs we will be writing, so we won’t bother, but when you write programs in the future, you may wish to use variables for columns also
* Only some of the program is shown below - the rest is just copy/paste



* You will notice the first formula is **row\_number = row\_number + 1**
* Then it changes to **row\_number +=1**
* The second is the Python short version of incrementing a variable
* There will be times when you are doing column 1 and column 2 separately that you will need to reset row\_number back to a lower variable so it all matches up
* You should do this by subtracting a constant, not just setting the variable to a lower number as it increases flexibility



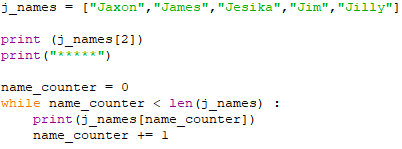
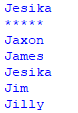
* Doing it this way means that if you need to add bracing timber you would just increase the constant from 7 to 8. This type of flexibility is required for better grades
* Of course, you will also have clear comments explaining what it does even at Achieved level

Task time

* Make whatever changes are needed to your program to include this constant

# reviewing lists

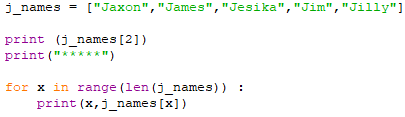
* Lists were covered at level 1. At level 2 we need to take them further
* Lists can be used to store multiple sets of data - they can also be added to and edited

* List items start a 0 (zero)
* Lists use square brackets to separate items
* You can print out a single item in a list by putting in the number the item is in the list
* A while loop and a counter can be used to print out all items from a list
* Instead of putting in **5**, which is the number of items in the list **len(j\_names)** is used. This returns the value **5**. It is a more reliable and flexible method of doing it

## The for loop

* Instead of using **while** you can use **for** instead - it is a little simpler
* The only with for loops is you use **x** instead of a well named variable, but it is following standards
* Printing **x** has only been included as a demonstration you probably not normally bother

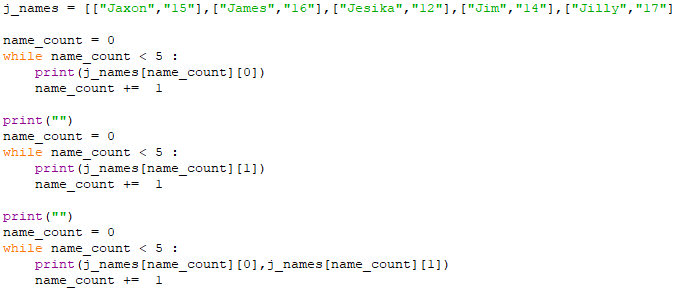
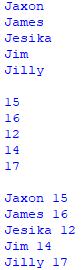
 

Task time

* Print off a list of 6 of your fellow students
* Add one more student to the list and repeat
* Don’t forget to add comments

# about mulidimensional lists

* When we get back to using a GUI we will find nested lists very useful
* Each list item is a separate row and each nested item is a separate column in that row

Task time

* Add another name to the list
* Add another column for male or female and print (feel free to change the names if you wish)