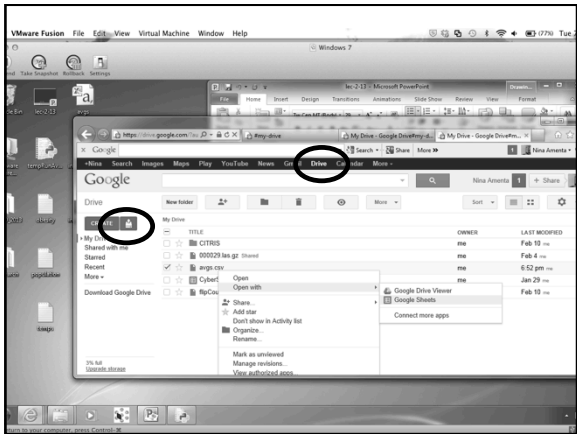
ECS 10

2/13

Announcements

- Assignment due Thursday.
- What to hand in
  - Program
  - Chart (in any image format)
- One way to get chart:
  - Upload file to Google Drive
  - Open with…Google sheets
  - Insert chart; pick line chart; Use column A as labels.



Function example - cube

```
def cube(nLocal):  # the input in parens
        nOut = nLocal*nLocal*nLocal
        return nOut  # the output

def main(): # no input
        x = 10
        xCubed = cube(x)
        print(x,"cubed is",xCubed)

main()
```

Local variables

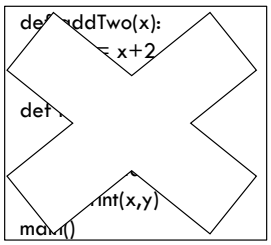
- Each function is like it's own little program
- The variables in cube() are only defined inside cube().
- The variables in cube() undefined in main()
- The variables in main() are undefined in cube()
- You ***could*** give them the same name, but it would lead to confusion. Two different variables should have two different names.

Example with same names

```
def addTwo(x):
        x = x+2
        return x
def main():
        x = 7
        y = addTwo(x)
        print(x,y)
main()
```

- After this runs, what's x? What's y?

Example with same names



Bad idea

- After this runs, what's x? What's y?

The Rule

- If you're using functions, no code outside the main() function.

```
def main():
        ….
main()
```

NOT ALLOWED

```
def addTwo(x):
        x = x+2
        return x
def main():
        x = 4
        y = addTwo(x)
        print(x,y)
x = 7
main()
```

Why?

- This makes the behavior of variables pretty simple; every function has its own local variables.
- The x in one function is a different variable from the x in another function.

Order of stuff in program

- Imports first.
- Next, all function definitions, with main() last.
- At the very end, just one statement outside a function:

```
main()
```

One way in, one way out

```
def addTwo(z):
        z = z+2
        return z
def main():
        x = 7
        y = addTwo(x)
main()
```

- Input to a function is through its parameters (can be lots).
- main() passes data to addTwo().
- x is the argument of addTwo() in main()
- z is the parameter of addTwo()

One way in, one way out

```
def addTwo(z):
        z = z+2
        return z
def main():
        x = 7
        y = addTwo(x)
main()
```

- z is the return value of addTwo()
- Output is only through its return value (only one).

Benefits of The Rule

- Variables in a function are local to that function.
- Information is passed between functions using arguments/parameters and return values.
- The flow of information should be clear.
- The main function is a roadmap to the rest of the program.