9/4 CSP(L) notes

### **Overview**

**idk**

### 

* **QOTD 1 is due before friday 8am**
* **It is unusual, its a survey assessing experiences attitudes and perspective on generative AI**
* **Completion grade**
* **Qotd 2 will be released friday and due monday 8am**

**Functions:**

* Function invocation itself “stands in” for a value
* Fundamental to the notion of programming is the ability to extend the language by writing your own function
* ‘Def’ is used to initiate a function definition. ‘Def frame ( parameters): <body of definition indented> —-- return( <return values if any>)
* End of function based of indentation
* Also: functions typically return values to use in future calculations.
* ‘Def frame(param1, param2, … paramN):
* Def addemup(x, y): …. return (x + y)
* Whatever arguments are given at invocation are evaluated and then passed to the function, mapping them 1:1 to the corresponding formal parameters
* >>> addemup(3, 4),,,, 7
* When u define, the formal parameter must stay same

Python prefers r and b to have same sign in // and % division

* Numeric types: **integer, floating points,** complex
* Logic types: **Boolean.**
* Sequence types**: string,** byte, bytearray, **list, tuple,** and **range**
* Mapping types**: dict.**
* Set types**: set,** frozenset
* Also: objects, functions, methods, modules, classes and others

Integer divided by floating point = floating point: mixed operations convert integers to float when floats are involved

Define function kg2lb that converts value given in kilograms to its equivalent value

* def kg2lb (w):
  + Return (w \* 2.2)
* Kg2lb (1)..... 2.2
* Kg2lb (3.7) ….. 8.14

Boolean:

* True and False
* Not False… True
* False and True… False

We can ask python REPL to reveal the type of an element

>>> type(3) .. class ‘int’

>>> type(17.0) class “loat’>

type(False)... Boolean

* You can convert elements between types
* >>> int(False) … 0
* >>> bool(1)... True
* >>> float (7+ (2-3)) … 6.0
* >>> float(not False)... 1.0