COMPUTER SCIENCE

ESSON 20+21, TUESDAY NOV 21st + THURSDAY NOV 23^{re}

WWW.CITA.UTORONTO.CA/~WOODFORD

WOODFORD@CITA.UTORONTO.CA

 \bigcirc

THIS WEEK IN CS AND STEM

- IBM announces 50 qubit computer
 - https://futurism.com/ibm-announced-50-gubit-guantum-computer/
- 2nd closest exoplanet ever discovered and it's looking pretty temperate
 - <u>https://futurism.com/discovered-closest-earth-like-planets/</u>



UNIT TESTING SCRIPTS

- We learned a lot about testing last week let's streamline our process.
- It was mentioned that we can put our unit tests into a single script and run them all at the same time. Let's try this for Lecture 18-19_UnitTesting.py and ...2.py.
 - What would such a script look like? How would it identify fails and successes? What would it output? Etc.
 - Note that Lecture 18-19_UnitTesting.py works as it is so your unit testing script should come back with successes. Try putting in an error in one of the functions to ensure your script is working!

PRECONDITION AND POSTCONDITION: REVIEW

- The main advantages are that they succinctly describe the behaviour of a function without cluttering up your or the users thinking of how the function actually works.
- Precondition = true before the function executres
- Postcondition = true after the function executes

USE THE PRE AND POST CONDITIONS TO MAKE A UNIT TESTING SCRIPT FOR OUR FREEFALL CODE

- Now we know how to identify starting and ending states, and how to unit test. Let's combine this knowledge to complete making pre and post condition statements and tests in Lecture 18+19_UnitTesting2.py and then make a unit testing script for this function.
- Remember the equations of motion:
 - $vf = vi + a^*t$
 - df = di + v^*t
 - t = t+dt (for each time step)

OTHER WAYS OF INCLUDING PRE AND POST CONDITIONS IN FUNCTIONS

• So far we've been using comments to include information about our functions and code. What are some cons of using comments?

OTHER WAYS OF INCLUDING PRE AND POST CONDITIONS IN FUNCTIONS

- So far we've been using comments to include information about our functions and code. What are some cons of using comments?
 - Not accessable by user unless they open the file
 - Can clutter code

• Usually you can instead include a docstring – let's try it out!

To print a docstring, you need to use : print(func.___doc___)

IDENTIFY COMMON PITFALLS IN RECURSIVE FUNCTIONS

• What is a recursive function?

RECURSIVE FUNCTIONS

- What is a recursive function?
 - A function that depends on previous calls of itself, or rather a result from a prior iteration.
 - Take the factorial function for example. You can either make it ITERATIVE or RECURSIVE
- How do recursive functions run? What does the implementation look like?
- When would you use a recursive function as opposed to an iterative one?

RECURSIVE FUNCTIONS

• What could be some pitfalls in recursive functions?

RECURSIVE FUNCTIONS

- What could be some pitfalls in recursive functions?
 - If the base case is missing (ie. the recursive function just keeps going forever in an infinite loop)
 - When the recursion step doesn't reduce to a smaller subproblem (ie. the program does not converge)
 - It takes too long, ie. much longer than the iterative function

WRITING RECURSIVE FUNCTIONS

- Many of you are already thinking recursively, so let's see if we can put together a recursive function for a function we already know how to write: a function for printing the Fibonacci sequence!
 - Iterative version:
 - Def Fibonacci_seq(n):

$$a,b = 0,1$$

for i in range(n):

print("a,b = "+repr(a) + repr(b))a,b = b,a+b

return

WRITING RECURSIVE FUNCTIONS

- Let's try a few more examples for recursive functions:
 - Greatest common divisor (gcd)
 - The sum of the first n natural numbers
 - Towers of Hanoi (a mind-game. It goes something like this: "There are three pegs which can hold stacks of disks of different diameters. A larger disk may never be stacked on top of a smaller. Starting with n disks on one peg, they must be moved to another peg one at a time. What is the smallest number of steps to move the stack?"
 - Hanoi(n) = 1 if n=1, 2*Hanoi(n-1)+1 if n>1



HOW COULD WE MAKE RECURSIVE FUNCTIONS MORE EFFICIENT?

- The main problem with recursive functions is that they tend to overcalculate or rather they calculate the same thing multiple times (think of the Fibonacci function!)
- What if we could hold the values and prevent the function from overcalculating? Can you think of a way to do that with the tools we already have?

DICTIONARIES

- This is a whole new kind of data structure called a set. It looks like a list, but there are some major differences.
- Dictionaries can be indexed by ANYTHING not just integers. These variables are called keys. It is also defined using curly braces, not brackets or parenthese.
 - Ex. Tel = {'jack':4098, 'sape':4139}

ASSIGNMENT #7

- Code/logic testing
- Follow the instructions in Assignment7.pdf. You can put your answers in a text, word, pdf, or pages document. Due Nov 26th by 11:59pm via email submission to <u>woodford@cita.utoronto.ca</u>

REFERENCES

- <u>https://docs.python.org/3/tutorial/controlflow.html#tut-docstrings</u>
- <u>http://web.mit.edu/6.005/www/fa15/classes/10-recursion/</u>
- <u>https://www.python-course.eu/recursive_functions.php</u>
- <u>https://docs.python.org/3/tutorial/datastructures.html</u>

UPCOMING LECTURES

- ROM Speaks Nov 21st : On Tyranny, 20 lessons from the 21st Century
 - https://www.rom.on.ca/en/whats-on/12th-annual-eva-holtby-lecture-on-contemporaryculture?utm_medium=email&utm_source=engagingnetworks&utm_campaign=holtby&utm content=12th+Annual+Eva+Holtby+Lecture+on+Contemporary+Culture

• ROM Speaks Dec 4th, 5th : Canada's Oceans, Towards 2020

- <u>https://www.rom.on.ca/en/whats-on/canadas-oceans-towards-</u> 2020?utm_medium=email&utm_source=engagingnetworks&utm_campaign=CanadaOce ans&utm_content=Canada+Oceans
- AstroTours Dec 7th : Gravitational Waves, Sirens of the Universe
 - http://www.astro.utoronto.ca/astrotours/?page_id=392