COMPUTER SCIENCE

LESSON 18+19, TUESDAY NOV 14TH + THURSDAY NOV 16TH

WWW.CITA.UTORONTO.CA/~WOODFORD

WOODFORD@CITA.UTORONTO.CA

THIS WEEK IN CS AND STEM

- Even further down the rabbit hole: An AI in Japan is granted residency.
  - https://futurism.com/artificial-intelligence-officially-granted-residency/
- Quark Fusion: the new nuclear fusion
  - https://futurism.com/quark-fusion-produces-eight-times-energy-nuclear-fusion/

ASSIGNMENT 5 SOLUTION

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 Lecture18-19_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 Lecture18-19_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!

WHERE TO FIND SUPPORTING DOCUMENTATION AND HOW TO USE IT

- Spyder: https://pythonhosted.org/spyder/index.html
- Jupyter: https://github.com/jupyter/help
- Python 3: https://www.python.org/ , https://docs.python.org/3/
- Other useful websites:
  - stack overflow (https://stackoverflow.com/questions/tagged/python)
  - Data Camp (https://www.datacamp.com/courses/tech:python, https://www.datacamp.com/courses/topic:machine_learning)

STARTING AND ENDING STATES, OR PRECONDITION AND POSTCONDITION

- What are these? Usually, before even writing your program, a programmer needs to communicate WHAT a function does without any mention of HOW.
  - When might this be useful? Can you think of an example?

STARTING AND ENDING STATES, OR PRECONDITION AND POSTCONDITION

- What are these? Usually, before even writing your program, a programmer needs to communicate WHAT a function does without any mention of HOW.
  - When might this be useful? Can you think of an example?

- Preconditions and postconditions are a pair of statements you can use to specify requirements for a function.
  - Precondition: what must be true BEFORE the function is called
  - Postcondition: what must be true when the function is FINISHED

PRECONDITIONS AND POSTCONDITIONS

- What might this look like in a function? Let's take a simple function as an example: the recursive factorial function. What would the pre and post conditions be for this function?

```
def factorial(x):
        if x ==0:
                return 1
        else:
                return x*factorial(x-1)
```

PRECONDITION AND POST CONDITION

- It's often good practise to write the pre and post conditions for a function as comments under the def line. It's not necessarily common practise however!
- What might happen if the precondition is violated? Who's responsible for ensuring the precondition is valid? Who's responsible for ensuring the postcondition is true at the end?

PRECONDITION AND POST CONDITION: QUIZ

- You call a function Yahweh(x,n) which takes x as a string and n as an integer. However, you pass x as a float instead. This causes a 40-day flood – who's responsible?
  - You
  - The programmer that wrote the function
  - Noah

PRECONDITION AND POST-CONDITION

- Even though it's ultimately the users responsibility to ensure the precondition is valid, the programmer can really make this easy or hard on them. What can you as the programmer do to help ensure the precondition is valid?

PRECONDITION AND POST-CONDITION

- Even though it's ultimately the users responsibility to ensure the precondition is valid, the programmer can really make this easy or hard on them. What can you as the programmer do to help ensure the precondition is valid?
  - Check for type/range/other indications that the precondition is not met
  - Print statements that state the precondition(s) before input
  - Well-documenting your code (comments, ReadMe)
  - If you detect that the precondition has been violated, halt the program and print an error message!

PRECONDITION AND POSTCONDITION

- Let's try writing these and checking for them on some functions that we're already familiar with – the sin, cos, tan, and factorial functions from Assignment 4 and that sit in Lecture18-19_UnitTesting.py. We can also try it out for Lecture18-19_UnitTesting2.py.

PRECONDITION AND POSTCONDITION: SUMMARY

- 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.
- It's also useful if you at some point reimplement the function in a new way but the pre and post conditions are still the same – making the actual usage and much of the documentation the same.

ASSIGNMENT #6

- Code/logic testing
- Follow the instructions in Assignment5.pdf. You can put your answers in a text, word, pdf, or pages document. Due Nov 19th by 11:59pm via email submission to woodford@cita.utoronto.ca

REFERENCES

- https://www.cs.colorado.edu/~main/supplements/chapt01.ppt
- http://www.cs.umd.edu/class/fall2002/cmsc214/Projects/P1/proj1.contract.html