CSC148H Week 12

Ilir Dema, Michael Miljanovic

Summer 2021

Communicating Through Recursion

Each time we call a recursive method:

- Everything it needs should be sent through parameters
- Everything it must report back should come through parameters or a return value
- Don’t attempt to work around this protocol by using local variables
- Each call has its own stack with its own separate local variables
- So nothing can accumulate in them across calls

Helper Methods

- Sometimes, a method’s API doesn’t “have enough” to support the recursion
- A helper can have an additional parameter. Example: The helper for Tree’s __str__ method
  - Adds a parameter for indent
- A helper can have an additional return value. Example: The helper for Tree’s average method
  - Returns a tuple with sum of values and number of values

What Did We Do?

You are no longer a novice programmer
You have written significant Python code and covered a broad range of practical experience, including:

- Working with code across multiple files and classes
- Using code-writing tools like PyCharm and python_ta
- Using testing and profiling tools (pytest, hypothesis, etc.)
- Learning useful Python libraries (e.g., pygame)

What Did We Do?...

Documentation

- We pester you to write good docstrings for a reason
- Communicate what your code does and how it is intended to be used
- What is assumed/implied?
- What is the expected result/outcome?
- NOT how it is done internally!
- Remember interface vs implementation!

What Did We Do?...

Debugging

- Follow the logic of your code, step by step
- Analyze if the behaviour of your code is as expected
- Important skill, especially for complex code
- Gets you out of tricky spots
- Calling print is fine sometimes, but don’t forget about debugging!

What Did We Do?...

Testing

- Important: test-as-you-go!
- Write meaningful tests
- This is a very important skill!
- Consider corner cases
- Your code must gracefully handle all exceptional cases instead of crashing
- Remember preconditions and representation invariants!
- Re-test your code on every revision
- Whenever you make substantial changes, re-test that everything that used to work still does

Code Aesthetics is Communication

“Programs must be written for people to read, and only incidentally for machines to execute.” — Harold Abelson

Documentation is Communication

“The most important single aspect of software development is to be clear about what you are trying to build.” — Bjarne Stroustrup

Software Design is Communication

“There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.” — Sir Charles Antony Richard Hoare

Exam Logistics

- 3 Hours, similar to Term Tests
- Everything in the course is fair game.

Some ideas for studying

- Practice in a test-like environment: at your computer, timed
- Look at past exams. Don’t ask for help/solutions too early!
- Form study groups! We can do this remotely.

Aim for Mastery

- *Mastery* means that you are the expert. Go beyond re-reading and re-doing
- Create multiple solutions
- Identify common mistakes or errors
- Explain ideas to others