CS1 Lecture 37 Apr. 19, 2017

- Exam 2
  - tomorrow April 20, 6:30-8:00pm, C20 Pomerantz Center
  - actual exam will be approximately 6:45-7:45pm
- NO CLASS FRIDAY, 4/21
- I will post Question 1 of HW 9 on Friday. HW 9 will have two questions and be due. Friday, April 28. Question 1 will be a small GUI/tkinter program. Q2 will be part 1 of program to be completed for HW10.

Last time

- One more problem solution using randomization – using Monte Carlo simulation to approximate π
- Introduction to GUI programming with Python

Today

- More GUI programming
- Review for exam 2

Last time: Graphical User Interface(GUI) Programming in Python

- tkinter is a commonly used Python layer on top of a standard GUI toolkit TCL/Tk
- GUIs built using *widgets:* basic building blocks including buttons, labels, text, entry fields, scroll bars, etc.
- Steps:
  - Define widgets and layout
  - Specify how to handle “events” (button presses, mouse clicks, etc.)
  - Start GUI/”event loop”

Getting started with tkinter

1. import tkinter
   - ***NOTE***: module was called Tkinter in Python 2. Much web documentation still calls is Tkinter. Functions, etc.. are the same but make sure to use 'tkinter' not 'Tkinter' when important module, calling functions, etc.
2. Before using any other tkinter functions, you must call Tk() to create root/main window and initialize Tkinter
   - e.g myMainWindow = Tk()
3. Next, define other GUI widgets – buttons, labels, entries, and their placement (via pack/grid function calls)
4. Specify how to handle event (e.g. button presses), typically by defining "callback functions" that are called by the GUI system when it detects events
5. Finally, when ready to start execution of the GUI, call mainloop()
   - e.g. myMainWindow.mainloop()

Note: mainloop() gives control of execution to Python. It won't return direct control to you until after you close/kill the main window. So, no commands/ function calls should follow it in your code. After you call it, you only get an opportunity to change things when *events* occur – i.e. when you click the mouse on buttons, type text in entries, etc.

- tkinter is big – there are long chapters and even whole books about it. You will use Tkinter in HW9 and HW 10 but not a lot of the features.
- You just need to understand the basics of a few widget types (Label, Entry, Button, Frame) and how to respond to events like button presses.
- Make sure you understand these four small examples (links available with Lec 36 on class web page)
  - minimal.py
  - minimal2.py
  - simplegui1.py
  - simplegui2.py

  and the example that you build upon in this week's discussion section

A bit on global variables since we will use them in GUI-based programs

- Previously discussed variable scoping. In functions, local variables are created for 1) formal parameters, and 2) any names that appear on left hand side of assignment
- What if you *want* to assign to top-level variable? You can use "global". Within a function

```
global x
```

  tells Python that a local variable should *not* be created for x. Instead, uses of x within the function refer to top-level variable x
- Examples: globalVars.py
- General advice: minimize use of global variables. Functions are easier to understand if they can be understand *locally*, without having to take into account a larger context (perhaps looking through and disentangling many files/functions). *However*, when carefully used they can be very useful.
- See "Global variables" section of Chapter 11 in textbook.

For HW 9 and HW 10 we will use several global variables to make it easy to share GUI-related and other values across multiple functions.

Links/resources for learning tkinter

- This tutorial - http://www.tutorialspoint.com/python/python_gui_programming.htm - does a good job of explaining and demonstrating the basics and includes several good small examples. It seems like a good place to start.
- If you want a "real textbook chapter", Chapter 6 of Kent Lee's book "Python Programming Fundamentals" is pretty good. The book is electronically available to UI students through the UI library (you need to be on the campus network to access it). http://link.springer.com/chapter/10.1007%2F978-1-84996-537-8_6
- This site can also very helpful. It's usually the first hit when I do Google search (though I'm not sure all info there is fully up-to-date): http://effbot.org/tkinterbook/tkinter-index.htm
- Tkinter info on the official Python site:
- https://docs.python.org/3.5/library/tkinter.html
- Some of the explanations here can be helpful (e.g. explains "pack" better than many others): http://thinkingtkinter.sourceforge.net

Exam 2 review

http://www.cs.uiowa.edu/~cremer/courses/cs1210/etc/exam2review.html (link is also on course web page)

Likely questions:

- dictionaries
- list comprehensions
- objects/classes
- Big-O
- binary search
- sorting
- graphs
- randomization/coin flipping