

- HW 9 scores posted
- DS 10 due this afternoon
- HW 10 due Saturday
- HW 11 will be posted tomorrow, due Sunday, May 9

Today

- Final course grade scales are on the next slide
- Continue with introduction to accessing web services such as Google Static Maps and Geocoding APIs
- Structure of HW10

Final Exam is optional. The default is that you are NOT taking it. You must OPT IN and notify me if you want to take it

Grade scales the same percentage-wise except for rounding differences

Without final

Possible points: 165

Grade	Points	approx %
A+	160	97
A	145	87.9
A-	139	84.2
B+	131	79.4
B	120	72.7
B-	112	67.9
C+	107	64.8
C	90	54.5
C-	82	49.7
D+	79	47.9
D	72	43.6
D-	66	40

With final

Possible points: 200

Grade	Points	approx %
A+	194	97
A	174	87
A-	166	84
B+	159	79.5
B	145	72.5
B-	135	67.5
C+	130	65
C	110	55
C-	100	50
D+	96	48
D	88	44
D-	80	40

IMPORTANT: To take the final, you **MUST** notify me by email by noon, Monday, May 10. If you sign up for the final you will be graded on the “with final” scale. You cannot choose the “without-final” scale after you opt-in for the final exam.

NOTE: POINTS ARE THE OFFICIAL SCALE – NOT PERCENTAGE

Student	Total	%	HW1	HW2	HW3	HW4	HW5	HW6	HW7	HW8	HW9	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS8x	DS9	Quiz 1	Quiz 2	Quiz 3	Quiz 4	HW	DS	Quizzes	Total	%	
Points Possible	156	100	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	15.00	20.00	20.00	20.00	54	27	75	156	100	
	152.00	97.44	6.00	6.00	6.00	6.00	6.00	6.00	6.00	5.00	5.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	14.00	20.00	20.00	19.00	52.00	27.00	73.00	152.00	97.44	
	147.00	94.23	6.00	5.00	4.00	5.00	6.00	6.00	6.00	6.00	6.00	3.00	3.00	3.00	2.00	3.00	2.00	3.00	3.00	3.00	3.00	14.00	20.00	20.00	18.00	50.00	25.00	72.00	147.00	94.23	
	137.00	87.82	5.00	6.00	4.00	5.00	6.00	4.00	5.00	6.00	5.00	3.00	2.00	2.00	3.00	3.00	2.00	3.00	3.00	3.00	3.00	14.00	16.00	20.00	17.00	46.00	24.00	67.00	137.00	87.82	
	121.00	77.56	5.00	5.00	4.00	4.00	5.00	6.00	5.00	5.00	4.00	3.00	3.00	2.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	14.00	20.00	15.00	43.00	26.00	52.00	121.00	77.56	
	112.00	71.79	5.00	5.00	3.00	0.00	0.00	2.00	6.00	6.00	4.00	3.00	3.00	3.00	2.00	3.00	3.00	3.00	0.00	0.00	3.00	14.00	19.00	14.00	11.00	31.00	23.00	58.00	112.00	71.79	
	112.00	71.79	4.00	6.00	4.00	0.00	6.00	5.00	0.00	6.00	1.00	3.00	1.00	3.00	0.00	1.00	3.00	2.00	3.00	0.00	3.00	11.00	20.00	17.00	13.00	32.00	19.00	61.00	112.00	71.79	
	105.00	67.31	4.00	5.00	3.00	2.00	4.00	6.00	4.00	6.00	5.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	0.00	3.00	3.00	8.00	11.00	12.00	11.00	39.00	24.00	42.00	105.00	67.31	
	105.00	67.31	5.00	6.00	5.00	3.00	6.00	2.00	5.00	4.00	4.00	3.00	3.00	0.00	2.00	3.00	0.00	0.00	3.00	0.00	3.00	15.00	20.00	13.00	0.00	40.00	17.00	48.00	105.00	67.31	
	105.00	67.31	5.00	5.00	3.00	0.00	6.00	5.00	0.00	1.00	3.00	3.00	2.00	2.00	3.00	0.00	2.00	3.00	0.00	0.00	3.00	7.00	17.00	17.00	18.00	28.00	18.00	59.00	105.00	67.31	
	102.00	65.38	4.00	6.00	2.00	2.00	3.00	2.00	6.00	2.00	4.00	3.00	3.00	3.00	3.00	2.00	0.00	3.00	2.00	3.00	3.00	14.00	13.00	9.00	13.00	31.00	22.00	49.00	102.00	65.38	
	99.00	63.46	2.00	5.00	4.00	4.00	4.00	4.00	4.00	3.00	4.00	3.00	2.00	2.00	2.00	2.00	3.00	3.00	3.00	0.00	3.00	8.00	12.00	7.00	15.00	34.00	23.00	42.00	99.00	63.46	
	91.00	58.33	5.00	5.00	5.00	2.00	6.00	5.00	4.00	2.00	0.00	3.00	3.00	0.00	2.00	0.00	2.00	3.00	1.00	0.00	3.00	6.00	8.00	16.00	10.00	34.00	17.00	40.00	91.00	58.33	
	87.00	55.77	3.00	4.00	2.00	3.00	4.00	5.00	3.00	3.00	0.00	3.00	2.00	2.00	2.00	2.00	3.00	2.00	1.00	3.00	3.00	10.00	8.00	10.00	12.00	27.00	20.00	40.00	87.00	55.77	
	84.00	53.85	4.00	4.00	0.00	3.00	4.00	5.00	4.00	5.00	4.00	3.00	3.00	2.00	2.00	2.00	2.00	3.00	3.00	3.00	3.00	6.00	4.00	8.00	10.00	33.00	23.00	28.00	84.00	53.85	
	80.00	51.28	3.00	5.00	0.00	2.00	4.00	2.00	5.00	3.00	2.00	2.00	3.00	2.00	2.00	2.00	2.00	2.00	1.00	3.00	3.00	8.00	6.00	10.00	11.00	26.00	19.00	35.00	80.00	51.28	
	46.00	29.49	4.00	3.00	0.00	0.00	4.00	2.00	0.00	0.00	2.00	3.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	5.00	5.00	8.00	8.00	15.00	5.00	26.00	46.00	29.49	
	19.00	12.18	3.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	2.00	2.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	3.00	4.00	0.00	0.00	4.00	8.00	7.00	19.00	12.18	

Structure of HW10start.py

- Classes
 - Globals. A class but we won't ever create instances. Just use the properties of the class as global variables. "Holding" them all in a Class object keep things a little bit better organized
 - properties include:
 - rootWindow
 - mapLabel
 - others you will add for new widgets you add to GUI
 - mapFileName, a string that never needs to change
 - mapSize, an integer that you can change if you want but doesn't need to change during execution of program
 - mapLocation, a string that you change via GUI
 - zoomLevel, an integer that you will change via GUI
 - other properties you will add such as mapType

Structure of HW10start.py

- Functions
 - `geocodeAddress(addressString)`
 - Calls Google geocoding service to get lat, lng for address specified in input string. Note: returns (0,0) if Google failed to yield lat/lng for the string
 - `getMapUrl()`
 - Uses values in Globals such as `mapLocation`, `zoomLevel`, etc. to construct a string that can be sent to Google to produce desired map.
 - `retrieveMapFromGoogle()`
 - Calls `getMapUrl` and then makes request to Google with that URL. Google returns an image that is stored in file `Globals.mapFile`
 - `displayMap()`
 - Calls `retrieveMap...` then then update GUI stuff to make map (which was stored in an image file by `retrieveMap...`) appear on `Globals.mapLabel` widget
 - `readEntryAndDisplayMap()`
 - **should** read string from Entry widget you add, save value in `Globals.mapLocation`, then display map. Initial version just displays map based on setting `Globals.maplocation` to "Beijing"
 - `initializeGUietc()`
 - Creates GUI, sets values of Globals properties so we can access GUI widgets in other code.
 - `HW10()`
 - initializes GUI, displays map, starts Tkinter loop

geocodeAddress function: using the Geocoding API from Python

Input: string for a location. Return value: a latitude, longitude pair

1. Create a URL string describing the geocoding info you want.
 - first few lines of geocodeAddress
2. Send the URL to Google and receive the results
 - urlopen(...) line of geocodeAddress
3. Google returns a JSON-formatted string.
 - JSON is a commonly used open standard for transmitting data as text. <http://en.wikipedia.org/wiki/JSON> Perhaps most commonly, JSON is used to transmit data in dictionary form – i.e. before transmitting, data is encoded into a long string that “looks” (to human reader) like a dictionary representation. The receiver of this string can then “decode” the string into a dictionary data structure and extract the items of interest out of it.
 - Json.loads(...) line decodes the JSON result, yielding a Python dictionary
 - Remaining lines look in the dictionary to extract the info we need (here, latitude and longitude). Study the Google Geocoding API - <https://developers.google.com/maps/documentation/geocoding/intro> to see what information is in the dictionary (much more than you’ll likely use!)

DEMO: change geocodeAddress to use Mapquest Geocoding API instead of Google. Only small changes needed. URL string changes and extracting lat/long from the JSON dictionary returned by Mapquest

Similarly, using the Static Maps service from Python

1. Create a long URL string describing the map you want. Use the documentation - <https://developers.google.com/maps/documentation/maps-static/intro> to learn the URL format
 - see getMapUrl in hw10start.py
 - You will need to modify getMapURL for the pin and maptype parts of HW10
2. Send the URL to Google and receive the results
 - see retrieveMapFromGoogle

HW10 todo list

- understand the use of class Globals as “nice” way of handling global values
- add Entry widget so you can change location
 - Upon button press, callback should read Entry, set mapLocation (global) variable, and call displayMap()
- enable zooming
 - Upon use of widget (button or whatever you choose - **consider +/-buttons as in simplegui2.py**), set zoomLevel (global) variable, call displayMap()
- enable changing map type
 - add a new global property for mapType
 - modify getMapURL to so that string representing request to Google Static maps API specifies desired map type
 - upon use of widget, set Globals.mapType value, and call displayMap() **Consider radio buttons as in radioButton.py**
- display pin at map center
 - modify getMapURL so that string representing request to Google Static maps API includes specification of pin at map center

- Demo: basic retrieving of Web pages via urlopen, etc. – `getWebPage.py`

Needed for HW11: DO SOON!

Definitely do step 1 before Monday

1. GET A TWITTER DEVELOPER ACCOUNT! Free

- <https://developer.twitter.com/>
- If you are worried about us seeing your personal tweets, make a different Twitter account just for this class
- It used to be simpler. Now you need to answer a bunch of questions. Just say you things like “For a university class project, learning about Twitter API and writing a Python program to search for tweets based on keywords and general location”. *The answers DO MATTER* – some students have had their requests denied based at least partly on these answers I think.
- For any required URLs (website, callback, etc.) when you are in the app creation screen, can just use <http://www.uiowa.edu> or similar
- *Some email addresses* (maybe particularly some international ones) *seem problematic* – result in delayed or denied approval.

2. Add required keys to **twitteraccess.py**. Test that searchTwitter() in twitteraccess.py works for you.

3. Then start working on HW11...

- More important than ever: do not write many lines of code before testing! *This assignment has a lot of code and many little things can go wrong*. If you add a lot of lines and then it crashes/doesn't work, it can be very difficult to debug/find where the error is.
- Add a few lines, test, add a few lines, test, ...

Next time

- Details and important information about HW 11
 - Twitter accounts
 - How to access/authenticate with web services like Twitter that require authorization