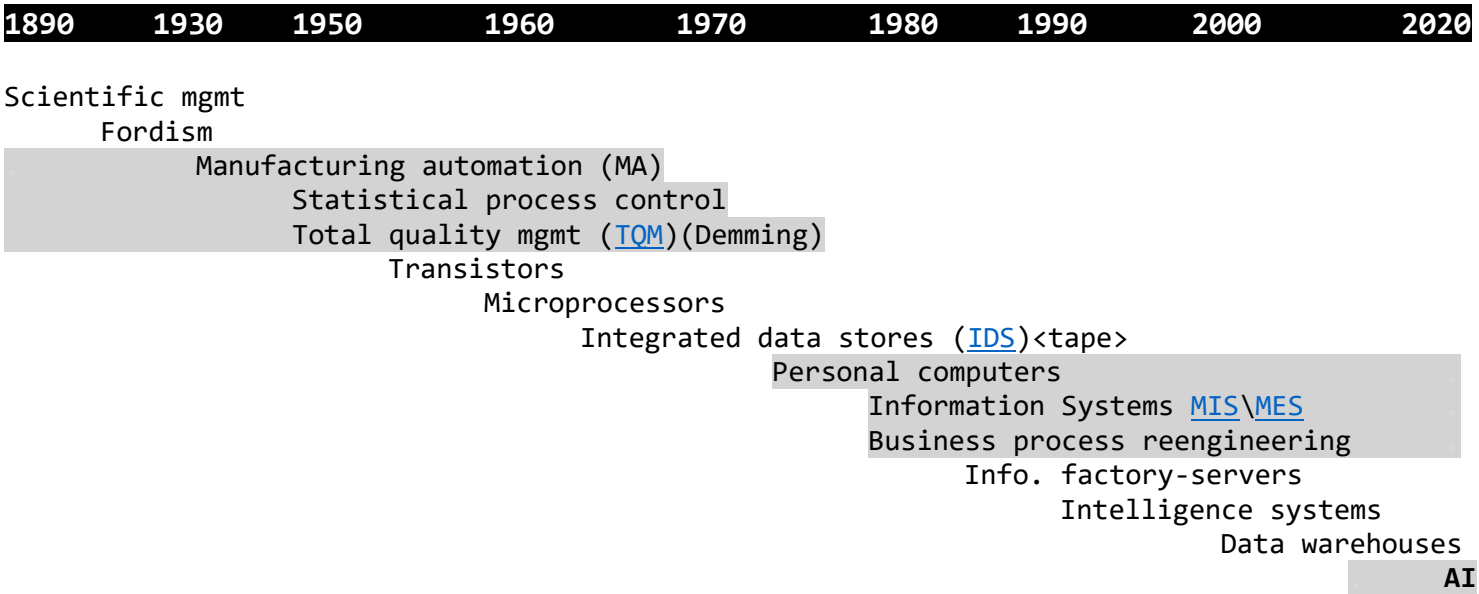
IT.304: Systems requirements and implementation planning
Course Prerequisites: IT.200, QSO.340
Location: on-ground, SETA, 209, Wednesday and Friday at 11:00 – 12:15
Instructor: brian hogan, b.hogan@snhu.edu, https://github.com/bbe2/IT.304.Fall.2022

Course Description

Systems analysis and design is an art form, discipline, and science. Thanks to Frederick Taylor, the 1890s embedded the formative pillars of speed, quality, and checklists as defined in his scientific management paradigm. American industrialization advanced from ratchets in the 1930s to silicon in the 1950s and information systems from the 1960s. Taylorism stopwatches, now timestamps, still measure efficiency through modern design and analysis methodologies[1].



To perform systems analysis and design well, it helps to know different reengineering models, the language of operation managers, what business leaders want to achieve, and critical impediments to sustainability. Information technology (IT) facilitates systems design efforts by codifying information. Today, Artificial intelligence (AI) helps drive system work by identifying unseen connections with tools like deep learning neural networks.

In the 1990s, MIT computer science professor Michael Hammer developed the management theory of business process re-engineering (**BPS**). Its tenets are process improvement, process re-design, and process re-engineering. BPS emphasizes the application of a holistic view to assessing how business objectives and processes are or are not aligned.

Question: have you stood in line in a coffee shop while the servers are busy doing many things but not helping you? IT online ordering has drastically improved systems, but great designs that don't measure or factor in an end-user or customer's needs results in rancid reviews and single stars. Quality system design principals seek to identify the internal and external factors a a priori to implement compelling holistic design experiences. versus implementing a poor customer design.

In the 2020s, **BPS** is alive and well, as witnessed by consultancies like IBM Business Process Reengineering and Bain's Business Process Redesign. **BPS** names change, such as Accenture Human + machine intelligence, but its Tayloristic principles remain exceedingly profitable.

Business requirements, rules, system specifications, environmental factors, opportunities to tear things apart, reorganize, recodify, and discover new viability vectors. **IT** is essential to this process. Understanding the application of BPS models will help you become a better system designer through the development of abstraction and looking-ahead skills. These skills improve with training and application.

In BPS, an individual's skills express as selecting, testing, and applying BPS models to frame situations. Abstracting systems involves using process engineering skills to help orchestrate quality engineered improvements, new IT paradigms, and machinery to augment and facilitate change. Measuring change is problematic and not a course focus. Suffice it to say; sometimes, only profit and stock price reflect the systemic effects of an organization's BPS efforts.

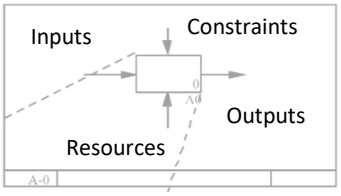
Why do BPS efforts wane? One answer is people and systems "move on." Life flows forward with designers and business champions refocusing and pulling the wind out of BPS's sails. Work not understood by managers leads to other ineffective, haphazard outcomes. Developing skills in this arena will help you identify concerns before a ghoulish nightmare.

This course will develop systems analysis\design skills as evidenced by,
1. Written examination and diagnostics of systems thinking.
2. Use of 10 modeling approaches to draft system requirements.
3. Use of object model programming to codify data and transactions paradigms.
4. Application of systems analysis and design principles by translating business and information structures into object models, systems requirement specifications, and.or implementation plans.
4. Evaluating a Harvard Business School case study (or similar).

The coursework is challenging, accessible, and extremely useful. As such, the expectation is your work will progress naturally in an ongoing fashion driven by self-interest and self-motivation. If a topic or assignment does not strike a nerve, please reach out to discuss it with the instructor.

Course competencies:

- IT-20358: Make ethically informed decisions based on awareness of legal and organization parameters
- IT-20359: Develop a systems requirements specification
- IT-30360: Develop an implementation plan



Required textbooks for knowledge reading assignments:
In any endeavor, resources are critical to success. In this course, information is assembled from various sources to minimize purchase costs. Printed materials are provided weekly alongside media stored in class bh.github. To the extent possible the instructor provides online references and only recommends quality materials. When applicable, consider acquiring materials from the SNHU Online Bookstore.

The following textbook is well suited for class purposes.

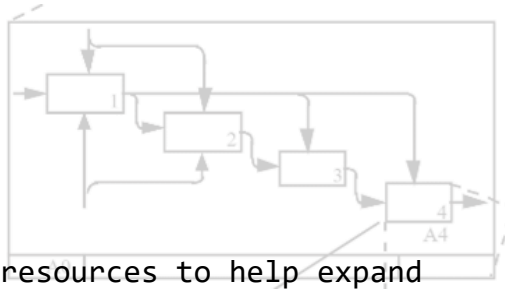
A) Tilley, Scott (2022). **Systems analysis and design, 12th Edition**. Shelley Cashman Series. Cengage. Published 2022. ISBN 978-0-357-11781-1.
- https://www.amazon.com/s?k=systems+analysis+and+design+12th+edition+scott+tilley&crid=3MA5XRRHG2KMB&sprefix=systems+analysis+%2Caps%2C82&ref=nb_sb_ss_ts-doa-p_2_17
- ***Instructor has 2 copies students may use and share for assignment readings.***

Models come in all forms. Ideas from The Decision Book will broaden your capabilities through fast weekly model exercises. Purchase encouraged.

B) Krogerus, M., Tschappeler, R., and Pienning, J. (2018). **The decision book: fifty models for strategic thinking.** ISBN-10: 0393652378, ISBM-13, 978-0393652376.
- Amazon.com: The Decision Book: Fifty Models for Strategic Thinking: 9780393652376: Krogerus, Mikael, Tschäppeler, Roman, Piening, Jenny: Books
-

Tools, technology, and software to facilitate evidence.

1. <provided> Paper, pencil, digital/in-hand Imperial rulers, index cards.
2. Document and spreadsheet software such as MS Word \ MS-Excel.
3. Microsoft Visio or another process design software like EdrawMax.
   ✓ Please attempt to submit .jpg or Adobe .pdf to help instructor consolidate work quickly.
4. Learn hands-on by applying weekly system models and theory to situations.
5. Blogs and disscussions chains can be integrated into the shared workspace.
6. Case studies to apply models too for assessment purposes.
7. A systems design and analysis custom model library at bh.github.
8. Software
   - Microsoft Visio (available through university here)
   - Python; jupyter notebook classic home
   - Python IDE: Jupyter :: Anaconda.org



note: students are not required to figure out code from scratch. They're expected to learn quality internet resources to help expand knowledge from provided codesets.

Instructor availability and response time.

- Communications typically occurs during class for the benefit of everyone.
- Interaction with the instructor and classmates will occur regularly on Wednesdays and Fridays at 11:00AM, room 209, in the SETA building.
- The instructor is available before and after class from 8 AM till approximately 3 PM for in person discussion. Please request a day ahead.
- Please communicate with your instructor via b.hogan@snhu.edu at any time!

A brief guide to effective analysis

This course is lecture based and taking notes is critical to both scholastic and business success.

i. In systems analysis and design, your interviewing customers to learn information and process details. Many people remain averse to recording conversations in any medium, so conversation recall is an essential skill. Augment your class notes shortly after a lecture to flesh out learnings, context, and details.

ii. Augment your class notes shortly after a lecture to flesh out learnings, context, and details. When something is unclear, please get in touch with your instructor promptly to help your analysis skills advance organically. Consider writing experience journals, as they are helpful to reflect upon if you seek employment in this field.

iii. **Blackout typing.** Consider typing your notes and ideas with a blacked-out computer screen. Doing so stimulates your abstraction engine flow.

iv. Word spelling/grammar matters. But, for now, focus on **IDEA** generation and design. The Victorians have 1000s of well-written texts nobody reads, and Herman Melville, a Victorian, wrote about a **process --** whale hunting.

v. Maintain a top 5 model list to facilitate and focus assignment work.

vi. Ask questions – right away. This course focuses on engineering ***courses of action***. Think ahead to clarify your thinking.

Good writing is good thinking.

Effective analysis items to do first:

1. Write down any ideas about the assignment and the models that come into your mind when they arrive. Carry index cards, text yourself, and keep a moleskin notebook and pencil. Don't put off recording something interesting for even five minutes, or else "whoosh-vapor."
    - Laboratory bench scientists still perform daily journaling of their activities.
    - It's worth considering forming this habit.

2. Carefully read every word of the assignment 2x to make sure you consider what lectures, readings, and models. Carefully cogitate an approach. Every assignment link is curated to deepen knowledge, focus thinking, **AND** eliminate internet research.
    - Consider reviewing the weekly assignment section and re-reading curated course content when your logic is amiss.
    - Between 4-8 will review strategies for librarian type research.

3. If an assignment indicates use of class lectures, ensure to study your lecture notes and materials provided.
    - Augment your lecture notes shortly after each lecture.
    - Consider adding notes in another color pen increase neuroplasticity.

Research Websites
The internet is full of information and advertisements. Use your time wisely working with the following quality research websites. The SNHU Shapiro library, ResearchGate, and Routledge will help provide most materials needed.

If you like what you find, consider setting up an account. Each provides unscheduled ad-hoc resource emails of quality information based on items you have queried.

Once acquainted with quality information sources, it is challenging to remember the data.Trash you likely wade through.

WARNING: when you appreciate quality information you may never listen to commercials again and use the internet quite differently.

Below are a few quality research website.

- Shapiro Library - Research Guides at Southern New Hampshire University (snhu.edu)
- Home Feed | ResearchGate, https://www.researchgate.net/
- Routledge - Publisher of Professional & Academic Books, https://www.routledge.com/
- Syracuse University Libraries – Research guides by subject
    - https://researchguides.library.syr.edu/
    - Syracuse also has outstanding librarians like MS. Brenna Helmstutler
        - https://researchguides.library.syr.edu/prf.php?account_id=152875

Note: Use of Wikipedia for course referencing and information sharing:

- Online dictionary Wikipedia can quickly inform on a topic's background providing a broad overview of topic's context and associated information.
- BUT, Wikipedia **is not** an academic reference nor a substitute for quality academic media. Some academics argue Wikipedia's veracity p.e.r.i.o.d.
- Submissions may not have a Wikipedia only reference so use sources provided.
- At any time a student may request academic approved learning media to substantiate any reviewed topic.

Diversity, Equity, and Inclusion
SNHU's core value indicates the university's commitment to "embrace diversity where we encourage and respect diverse identities, ideas, and perspectives by honoring difference, amplifying belonging, engaging civilly, and breaking down barriers to bring our mission to life."

Higher education work embraces the expansion and exhibit of growth mindsets. A growth mindset includes the practice of diversity, equity, and inclusion (DEI) to provide transformative experiences for yourself, peers, faculty, and staff. Our efforts form a collective, organized learning mechanism that helps ensure no one is left behind or alone in their learning experience. Through our community, compassion, and interactions, we walk in respect to the greater good possible in all of us.

SNHU Handbook and University General Guidelines

- https://snhu.sharepoint.com/sites/CAMPUSACADEMICS
- Use your internal resources to access the student handbook detailing all features of attendance, academic honesty et. cetera.
- Perform authentic work.
    - SNHU requires all students adhere to high standards of integrity including avoidance of plagiarism and cheating.
- SNHU adheres to copyright provisions of the Copyright Act.
- Consult the handbook when considering withdrawal or need anything else.

ADA/504 Compliance Statement

SNHU is dedicated to providing equal access to individuals with disabilities in accordance with Section 504 of the Rehabilitation Act of 1973 and with Title III of the Americans with Disabilities Act (ADA) of 1990, as amended by the American's with Disabilities Act Amendments Act (ADAAA) of 2008.

SNHU prohibits unlawful discrimination on the basis of disability and takes action to prevent such discrimination by providing reasonable accommodations to eligible individuals with disabilities. The university has adopted this policy to provide prompt and equitable resolution of complaints regarding any action prohibited by Section 504, the ADA, and the ADAAA.

For any questions about support services, documentation guidelines, general disability issues, or pregnancy accommodations please email wellness@snhu.edu. See my.snhu.edu and select the wellness tab. And the campus accessibility center at cac@snhu.edu.

For anything regarding discrimination please contact school professionals right away at the emails above and.or see the Disability and Accessibility Services at https:\\my.snhu.edu

Student Support Resources including Tutoring and Instructional Support

It is really amazing to have a careteam@snhu.edu to help students with assistance of all sorts. Again, this is an amazing resource.

- Consider this service if feeling pressured or overwhelmed.
- For instructional support email instructionalsupport@snhu.edu.
- For in class tech support call 603.645.9615

Other Key Resources

- https://snhu.sharepoint.com/sites/thesource
- https://snhu.sharepoint.com/sites/CAMPUSACADEMICS

Grading Guides

- Weekly activities and assignments are posted in this doc Friday evening to to bh.github.
- Specific category instructions, grading rubrics, directions, and hand-it-in requirements are detailed in the assignments.
- Grades and feedback are within seven days. This course also contains non-graded activities to assist you in mastering the learning outcomes.

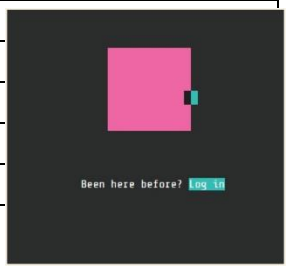
Grade distribution*

Category	# items	Points	Total points
Activities	10	60	600
Assessments/Quiz	5	20	100
Project 1	1	150	150
Project 2	1	150	150
		Total	1000

*based on class experience and expectations may be revised by 2nd Wednesday of week 2
**updated: 09.05.22, grading categories were finalized.

University grading system

Grade	Numerical Equivalent	Points
A	93-100	4
A-	90-92	3.67
B+	87-89	3.33
B	83-86	3
B-	80-82	2.67
C+	77-79	2.33
C	73-76	2
C-	a	1.67
D+	67-69	1.33
D	60-66	1
F	0-59	0
I	Incomplete	
IF	Incomplete/Failure	
IP	In progress	
W	Withdrawn	



Due Dates

Assignments are due anytime on the day of the world clock day. If it's December 31st "somewhere" an assignment is on time.

Weekly Assignment Schedule

Reading assignments, activities, and tasks are distributed at the end of week except for the first week on bh.github. For students interested in doing work ahead of schedule please contact instructor. The instructor advocates for courseload strain reduction to help ensure good thinking.

Templates

Wk	Focus & Medium	Weekly Topic & Assignment
x	~py pkg index~ https://pypi.org/	Hands - mediapipe (google.github.io) **note:** Weekly Assignments (1 or 2 pages per week as indicated on left)

Wk	Focus & Medium	Weekly Topic & Assignment
x		

Focus & Medium	Grow with Google Test Answer, b.hogan@snhu.edu

Program / operating parameters:
1. Demonstrate how to create a small Python program, called a script, and generate speech to text and text to audio results.
2. Challenge a user to replicate proper syntax, indenting, and other coding idioms to ensure programs run as intended.
3. Educate on basic data encoding where binary (1 or 0) is used for pictures/voice and nonbinary (byte/collations) is for text.
4. Educate on how libraries simplify program feature engineering making the art of the possible a far less daunting task.
 	
Scenario 1: Generate a working program in a Python integrated development environment (IDE) such as Anaconda. The following example uses the Jupyter notebook program as part of the Anaconda Install.
Scenario 2: Expand code requiring 2 audio requests but deliver a single audio outcome file
Hint: The trick of this scenario is to create 2 separate myWords variables.
• In Python variables are either implicitly or explicitly declared.
• Code line 7 “my Words” is an implicit declaration as its type is not declared, such a character (char) or number
	""" Part 1: Set Computer File Directory os=operating system"""
	import os
	os.chdir('C:\\Users\\17574\\Desktop')
	
	""" Part 2: Set Google Speech Recognition and Microphone Library Functions
	import speech_recognition as sr
	import pyaudio
	
	""" Part 3: Ask user to same something use Google speech to parse words"""
	with sr.Microphone() as source:
	  print("Ready? Say something quick")
	  myWords = sr.Recognizer().listen(source)
	  print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
	
	>>> Ready? Say something quick
	>>> You Said...: Nacho
	
	"""Part 4: Encode words into audio file audio data is binary so add 'wb'
	  for 'write binary data (1 or 0)"""
	with open("myAudio.wav", "wb") as file_:
	  file_.write(myWords.get_wav_data())
	
	"""Part 5: Import a generic microphone module """
	from playsound import playsound
	playsound('myAudio.wav')
	
	import os
	os.chdir('C:\\Users\\17574\\Desktop')
	import speech_recognition as sr
	import pyaudio
	with sr.Microphone() as source:
	  print("Ready? Say something quick")
	  myWords_1 = sr.Recognizer().listen(source)
	  print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
	with sr.Microphone() as source:
	  print("Ready? Say something quick")
	  myWords_2 = sr.Recognizer().listen(source)
	  print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
	myWords = myWords_1 + myWords_2
	with open("myAudio.wav", "wb") as file_:
	  file_.write(myWords.get_wav_data())
	from playsound import playsound
	playsound('myAudio.wav')
	
	>>> Ready? Say something quick
	>>> You Said...: Nacho
	>>> Ready? Say something quick
	>>> You Said...: Nacho
	
	""" Run like a Pro """
	import os
	os.chdir('C:\\Users\\17574\\Desktop')
	import speech_recognition as sr
	import pyaudio
	with sr.Microphone() as source:
	  print("Ready? Say something quick")
	  myWords = sr.Recognizer().listen(source)
	  print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
	with open("myAudio.wav", "wb") as file_:

• Add a "_1" to the variable and then duplicate code lines 5-8
with a second variable myWords_2
• Finally, combine the myWords_1 with myWords_2 into myWords
to deliver the audio output
	`    file_.write(myWords.get_wav_data())`
	`from playsound import playsound`
	`playsound('myAudio.wav')`
	
	`>>> Ready? Say something quick`
	`>>> You Said...: I like cake`

Wk	Focus & Medium	Weekly Topic & Assignment
16 **Dec** **12** **to** **17**	**Final Week !** **Machine Learning** Science fiction grand master and inventor of grok Robert Heinlein  Robert Heinlein at the 1976 World Science Fiction Convention in Kansas City, Kansas	Machine Learning faciliated the advancement of artifiicial intellence It is IT.304.Fall.2022/xwk_16_ML_Syllabus_1wk_intensive.pdf at main · bbe2/IT.304.Fall.2022 (github.com) Google Josh Gordon Cheatsheet IT.304.Fall.2022/xwk_16_(machineLearn)_josh_gordon_list.pdf at main · bbe2/IT.304.Fall.2022 (github.com)  Inspiration to be a Unicorn distinguish yourself by activites and skills IT.304.Fall.2022/xwk_16_(advice)_Be_a_Unicorn_(reddit).pdf at main · bbe2/IT.304.Fall.2022 (github.com)

In closing:

I. get your grok on
II. believe in yourself!
    a. claude m. bristol, the magic of believing. the classic guide to unlocking the power of your mind
III. good writing is good thinking
IV. water cooler talk = nothing personal but share something consistent
    b. solid strategy is talking about the latest book your reading, podcast, youtube.
V. skill to upskill (and learn to love it)

From an anomymous high-level computer science executive. Consider reading it.

- be a unicorn
- distinguish yourself with skills + activities
- never be bored

IT.304.Fall.2022/xwk 16 (advice) Be a Unicorn (reddit).pdf at main · bbe2/IT.304.Fall.2022 (github.com)

Focus & Medium	Grow with Google Test Answer, b.hogan@snhu.edu

wk.16 Machine Learning 12.12-12.16

Program / operating parameters:

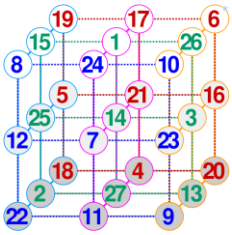
1. Demonstrate how to create a small Python program, called a script, and generate speech to text and text to audio results.
2. Challenge a user to replicate proper syntax, indenting, and other coding idioms to ensure programs run as intended.
3. Educate on basic data encoding where binary (1 or 0) is used for pictures/voice and nonbinary (byte/collations) is for text.
4. Educate on how libraries simplify program feature engineering making the art of the possible a far less daunting task.

Scenario 1: Generate a working program in a Python integrated development environment (IDE) such as Anaconda. The following example uses the Jupyter notebook program as part of the Anaconda Install.

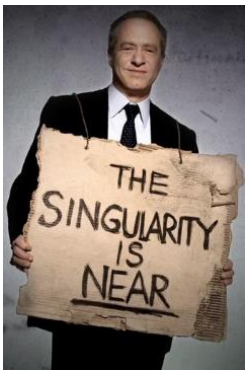
Scenario 2: Expand code requiring 2 audio requests but deliver a single audio outcome file

```
""" Part 1: Set Computer File Directory os=operating system"""
import os
os.chdir('C:\\Users\\17574\\Desktop')

""" Part 2: Set Google Speech Recognition and Microphone Library Functions
import speech_recognition as sr
import pyaudio

""" Part 3: Ask user to same something use Google speech to parse words"""
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))

>>> Ready? Say something quick
>>> You Said...: Nacho

"""Part 4: Encode words into audio file audio data is binary so add 'wb'
                            for 'write binary data (1 or 0)"""
with open("myAudio.wav", "wb") as file_:
    file_.write(myWords.get_wav_data())

"""Part 5: Import a generic microphone module """
from playsound import playsound
playsound('myAudio.wav')

import os
os.chdir('C:\\Users\\17574\\Desktop')
import speech_recognition as sr
import pyaudio
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords_1 = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords_2 = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
myWords = myWords_1 + myWords_2
with open("myAudio.wav", "wb") as file_:
    file_.write(myWords.get_wav_data())
from playsound import playsound
playsound('myAudio.wav')
```

Hint: The trick of this scenario is to create 2 separate myWords variables.
• In Python variables are either implicitly or explicitly declared.
• Code line 7 “my Words” is an implicit declaration as its type is not declared, such a character (char) or number
• Add a “_1” to the variable and then duplicate code lines 5-8 with a second variable myWords_2
• Finally, combine the myWords_1 with myWords_2 into myWords to deliver the audio output
	>>> Ready? Say something quick
	>>> You Said...: Nacho
	>>> Ready? Say something quick
	>>> You Said...: Nacho
	
	""" Run like a Pro """
	import os
	os.chdir('C:\\Users\\17574\\Desktop')
	import speech_recognition as sr
	import pyaudio
	with sr.Microphone() as source:
	print("Ready? Say something quick")
	myWords = sr.Recognizer().listen(source)
	print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
	with open("myAudio.wav", "wb") as file_:
	file_.write(myWords.get_wav_data())
	from playsound import playsound
	playsound('myAudio.wav')
	
	>>> Ready? Say something quick
	>>> You Said...: I like cake

Wk 15 Dec 5 to 10	Weekly Topic & Assignment
On Day.2, the inclass lecture will review course materials so you can complete the following confidently. **Final Essay 1 of 2**   3x3x3 magic cube with rows summing to 42 	**Final Essay** **Instructions**: Use materials from lectures, notes, and anything and everything on the class GitHub. The more content you use, the better. The goal is to ensure you understand the essence of systems analysis, design tools, methods, and thinking. Ideas and content are more important than perfect grammar, but good writing is still good thinking. Hand drawings are fine when done with care. **essay.1** "GrugooUck?" huh? When you agreed to work for Commander Lambda, you knew your life would be different, but no one realized it would be like living with comedic actor John Belushi as a re-incarnated Slimer. Besides being utterly phantasmagorical, the real issue is the office stench from Slimer residue. Slimer minions are ultra-elite workers but are also very opinionated. They will do any, and all work asked of them under any circumstances. Who wouldn't want this team? However! If all 42 agree on something, YOU must make a change and do so immediately. You need a solution by noon, or you will freeze in carbonite for an unknown time. The issue is email. All 42 minions have decided it's garbage. There's incessant quibbling about the barrage of this foul form of human communication. The Slimers voted! They have had it to their pits! They squack it's senseless, not process-oriented, and half the time full of jibberish. The Chief Slimer, Doug, says it oozes crapper garbage, and why should they decipher whether something is requested or needs to get done? He also gurgles something about IBM having tools for this. For instance, his last email was about Outpost_7 requiring 88 grenades to resupply from yesterday's

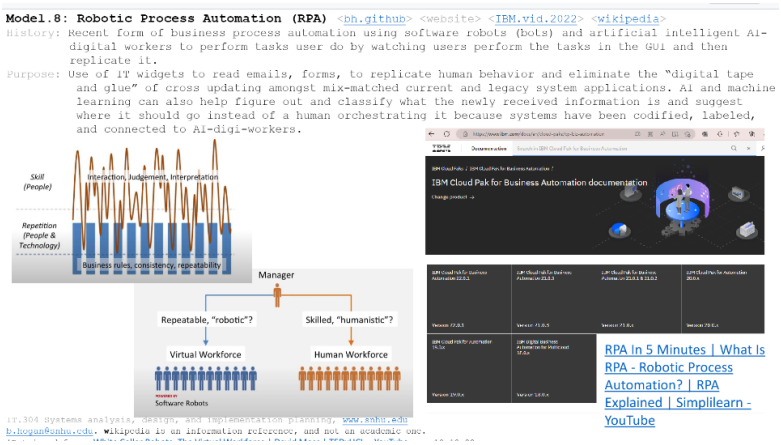
	arachnid attack. That's it! Why is an email like this necessary? Doesn't it seem wicked redundant and inefficient? Your actions: Describe a system to replace email and diagram as needed. Ensure the discussion includes what information will go to which Python data object so the propeller heads can do the programming. Ensure to consider data fields to, from, subject, and content. Also, consider the super weird minion's "Ah-sooo" idea about using natural language processing and understanding AI to parse text into performing actions, requesting actions, and no action categories. Ah-sooo is so happy you're getting busy with this, as he feared your carboniting would maim you. For whatever reason, he touches you and says, "gurgooUck."

Wk 15 Dec 5 to 10	Weekly Topic & Assignment
Final Essay 2 of 2  	Congratulations! You started today as a Google analyst working for legendary Ray Kurzweil. You parked successfully and got into the building by an eye scan? What's super strange is everyone is wearing headphones, and it doesn't seem like anyone will start talking with you. Maybe they would, but you're afraid to ask. Ex Machina finds you! Whew. She starts a long conversation, asking you everything you need to be comfortable, happy, and at peace with yourself. After 20 minutes, she even asks if you need quiet time to sit down and relax. The whole thing has been confusing, but your senses immediately kick into overdrive! Are you being tricked by Ray somehow? Is this a weird Jedi feeling thing? You figured out a solution and returned to your training and skills asking, how am I going to do something interesting? You look at Machina and ask her "any chance I can have some headphones?" Curiously, she smiles and says "sure – see you later googler" and leaves. She drops a piece of paper in front of you, and you're puzzled if this is a Ray test or something else. 01010111 01100101 01101100 01100011 01101111 01101101 01100101 00100000 01100111 01101111 01101111 01100111 01101100 01100101 01110010 00100001 00100000 01001001 00100000 01101110 01100101 01100101 01100100 00100000 01110100 01101000 01110010 01100101 01100101 00100000 01100110 01100101 01100001 01110100 01110101 01110010 01100101 01110011 00100000 01100100 01100101 01110100 01100001 01101001 01101100 01100101 01100100 00100000 01100110 01110010 01101111 01101101 00100000 01110100 01101000 01101001 01110011 00100000 01100010 01111001 00100000 01101110 01101111 01101111 01101110 00101110 00100000 01001111 01101110 01100011 01100101 00100000 01100100 01101111 01101110 01100101 00101100 00100000 01100110 01101001 01100111 01110101 01110010 01100101 00100000 01101111 01110101 01110100 00100000 01101000 01101111 01110111 00100000 01110100 01101111 00100000 01110011 01100101 01101110 01100100 00100000 01101001 01110100 00100000 01110100 01101111 00100000 01101101 01100101 00100000 01110011 01101001 01101110 01100011 01100101 00100000 01110111 01100101 00100000 01100100 01101111 01101110 00100111 01110100 00100000 01101000 01100001 01110110 01100101 00100000 01100101 01101101 01100001 01101001 01101100 00101100 00100000 01100010 01110101 01110100 00100000 01111001 01101111 01110101 00100000 01100011 01100001 01101110 00100000 01100001 01110011 01101011 00100000 01001101 01101111 01101101 01100001 00100000 01100001 01101110 01100100 00100000 01110100 01101000 01100101 01101110 00100000 01110011 01110000 01100101 01101110 01100100 00100000 01110100 01101000 01100101 00100000 01100001 01100110 01110100 01100101 01110010 01101110 01101111 01101111 01101110 00100000 01110100 01110010 01111001 01101001 01101110 01100111 00100000 01110100 01101111 00100000 01100110

Hint:
A second binary desciription is on this page but it is hidden, not encoded. Use your resources.

01101001 01101110 01100100 00100000 01101101 01100101 00101110 00100000 01010101 01110011 01100101
00100000 01110100 01101000 01100101 00100000 01100001 01100110 01110100 01100101 01110010 01101110
01101111 01101111 01101110 00100000 01110100 01101111 00100000 01101100 01101111 01100011 01100001
01110100 01100101 00100000 01111001 01101111 01110101 01110010 00100000 01100101 01100111 01100111
00100000 01110000 01101111 01100100 00100000 01100001 01101110 01100100 00100000 01101101 01100101
00101110 00100000 01010100 01101000 01100001 01101110 01101011 01110011 00101100 00100000 01110010
01100001 01111001

工具	kütüphane	Games
DataMAN 计算器 скользящая линейка	μαθηματικά טראַפ טראַפ 熊貓	Guess a number Verify answers New inch ruler game

Wk	Focus & Medium	Weekly Topic & Assignment
14 **Nov 28 to Dec 2**	**Objective** **Discussion**	foobar: read://https_www.ibtimes.co.uk/?url=https%3A%2F%2Fwww.ibtimes.co.uk%2Fgoogle-foobar-how-searching-web-earned-software-graduate-job-google-1517284 Julie chiles: Tarts & Pies with Leslie Mackie | Baking With Julia Season 2 | Julia Child - YouTube https://www.youtube.com/watch?v=ulD728JBKf8
13 **Nov 21 to 26**	**Robotic Process Automation Model**	https://www.youtube.com/watch?v=ulD728JBKf8  ➢ IT.304.Fall.2022/Model_8_RPA (robotic process automation).pdf at main · bbe2/IT.304.Fall.2022 (github.com)

Desert Type	Dough formula
1] Tart 2] cake 3] pies 4] breads 5]	1] 2] 3] 4] 5]

Ingredents	Bakery	Time
1] pastry flour 2] salt 3] shortening 4] 5]	1] 2] 3] 4] 5]	1] 2] 3] 4] 5]

Listen up rumpSwaps. I adopted earth's galting gun to yoru email bullet idea your email bullet idea. Your going to love it!

Each minion now gets their own "rotating mammy." You come to work and the load manny by belting out the message bullets.

Each bullet has their own container, message action, and and GO button.

Once belted up, embrace the suck and let ur rip.

then collect the chinks and containers filing chinks to trash and containers to archive

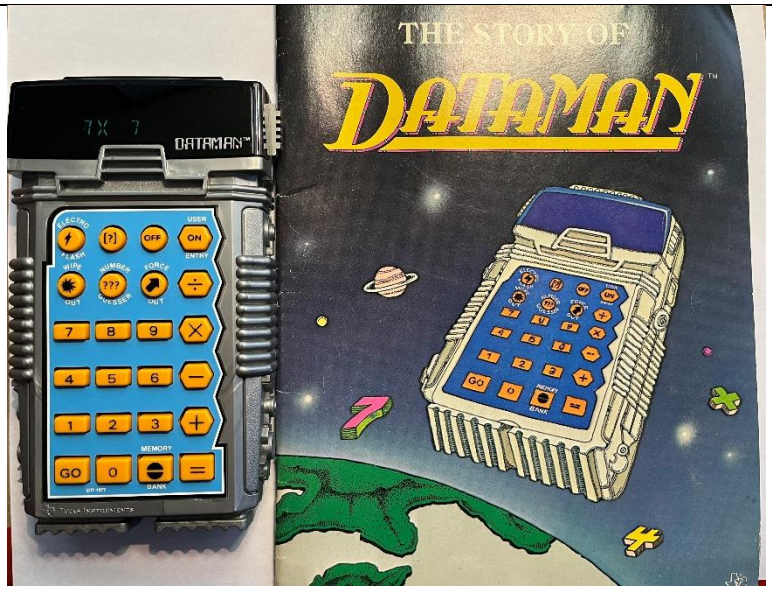
get going rumpswaps ~DD

- You will each get an electronic LED screen and you will shot the emails at.
- their self guidance system will deliver the "message action" with the details to complete the incoming request

- Email bullet converion tool
- Emails parsed into electronic shells <see data diagram>
- Your deliver

Wk	Focus & Medium	Weekly Topic & Assignment
12 **Nov 14 to 19**	**Objective** **Discussion** **Assignment: Readings** **In-class Discussion**	DataMan_US.pdf (datamath.net)

Questions



https://patents.google.com/patent/US4340374A/en <dataman patten>
inventor bobby g culley

Sean Riddle's Home Page - TMS-1980; https://seanriddle.com/tms1980.html <source code for the device>
hexadecimal

```
digits
   -       -       .       -       -       -       -       -       - -
  |   |   |   |   \ | /   |   |   |   |   -   |   |   |   |   |   |   |     |
   -       -       -       -       -       -       -       -
  |   |   |   |   / | \   |   |   |   |   -   |   |   |   |   |   |   |     |
   -       -       .       -       -       -       -       -       - -
   9       8       7       6       5       4       3       2       1

segments
   A       A       A       A       A       A       A       A       F A
 F   B   F   B   F B F   F   B   F   B   =   F   B   F   B   F   B   G     B
   G       G       G       G       G       G       G       G
 E   C   E   C   F E F   E   C   E   C   =   E   C   E   C   E   C   E     =
   D       D       D       D       D       D       D       D       D C
```

programmers reference manual for the tms1000 series
https://en.wikichip.org/w/images/f/ff/TMS1000_Series_Programmer%27s_reference_manual.pdf
from page 2-2

> **2-2 ROM ADDRESSING.**
> The ROM has 8,192 possible matrix points (1024 eight-bit words) where MOS transistors are placed to define the bit patterns of the machine language code. The ROM is organized into 16 pages of 64 words each (16 x 64 = 1024 words total). Each word contains eight bits.

http://www.seanriddle.com/mp3438a_rom_acid.jpg

assembler language, C, C+

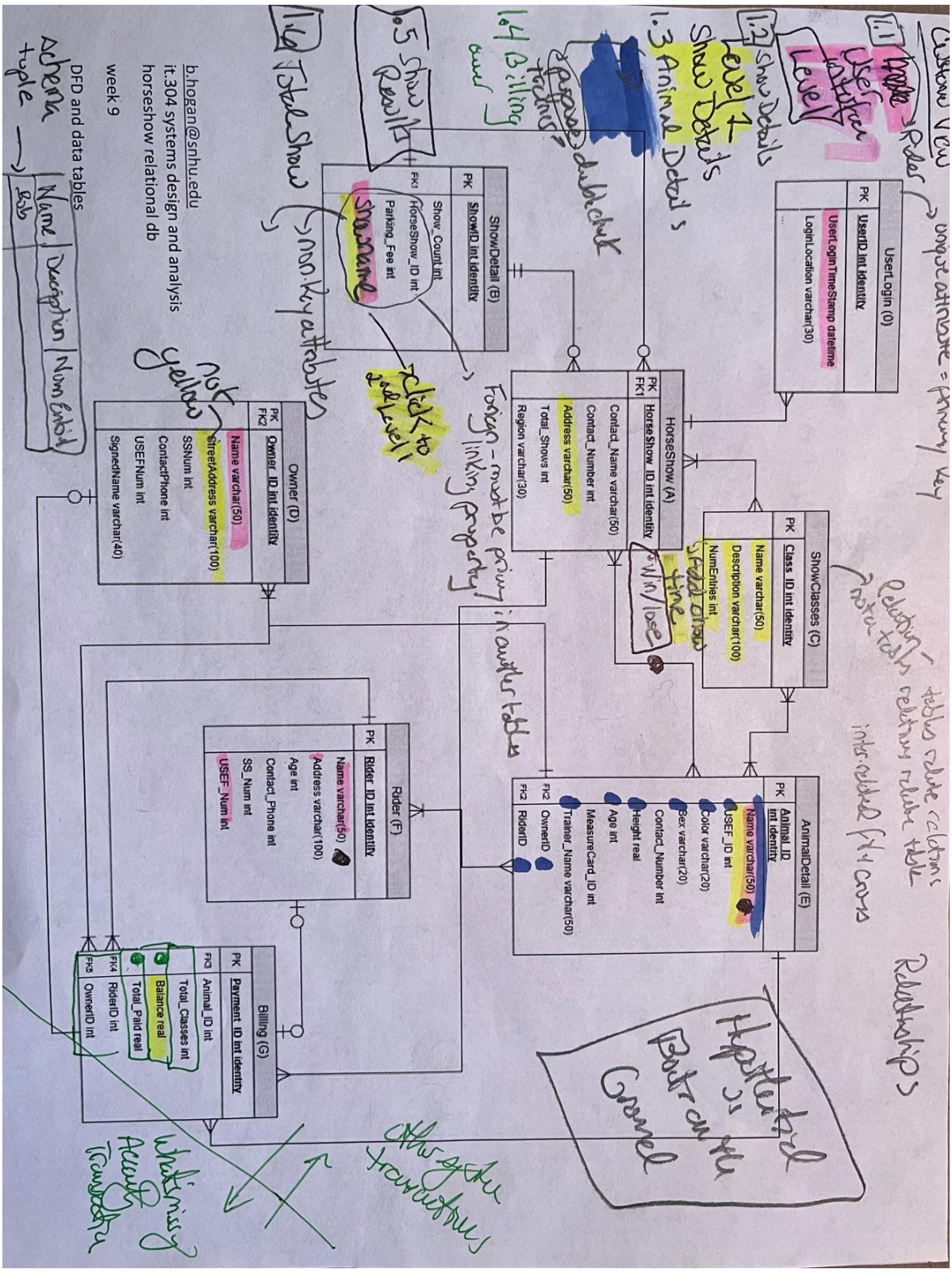
http://www.datamath.org/index.htm
joerg@datamath.org

http://www.datamath.net/Manuals/DataMan_US.pdf (source)

user manual:
https://github.com/bbe2/IT.304.Fall.2022/blob/main/xwk_12_DataMan_UserManual.pdf

article:
https://github.com/bbe2/IT.304.Fall.2022/blob/main/xwk_12_Dataman_Article_Dyson_J_Dec_2021.pdf

Wk	Focus & Medium	Weekly Topic & Assignment
11 **Nov 7 to**	Lecture review: Introduced Data Flow Diagrams (model.4) as a primary means to perform system design. You can quickly map data to potential application features by working with the key data fields. In this example, we used the HorseShow diagram and five features a horse rider would need at a horse show and used color to map the data by feature. Some designers start thinking about "what is nice to have" or theoretical. But, if you get right to work on what you can make happen, you can reduce aggravation and drive conversations with customers on what is essential.	

12	

Wk 10	Focus & Medium Oct 31 to Nov 5	>_7.Pillars.of.Python

```
it.304 week 10 EXAM - 7 pillars of python, Created on Sat Oct 22 07:42:57 2022, Updated 10/31/22
#=> Good writing is good thinking, AND good programming is mad-hatting
#=> due date =  when you thoughtfully complete the exercises
link to materials => https://github.com/bbe2/IT.304.Fall.2022/blob/main/code_wk_10_project_code_workbook.py
```

Over the past few weeks, have you taken the time to learn Python data structures, data pack/unpack, data transformation, conditionals, iterators, functions, and building object-oriented classes. Whew!

This work is a significant undertaking, and all autobots.it304 have performed amazingly well. You have shown your capacity to think differently and "sticktoitiveness." You are doing the work others find challenging.

Now let's take Shakespeare from the bottom up. In this challenge, you will work directly with text files and duplicate the spreadsheet tables from weeks 5-8. You will import, transform, iterate, use conditionals, and end with a simple object asking user what they want to read and giving it to them. Use your skills and make it so. ~brianh

p.s. Please provide quality and thoughtful answers to all questions. Please write as much as you want, ask me questions, and use this as your opportunity to build a piece of evidence to showcase an employer. I am happy to help you expand items if you would like to do.

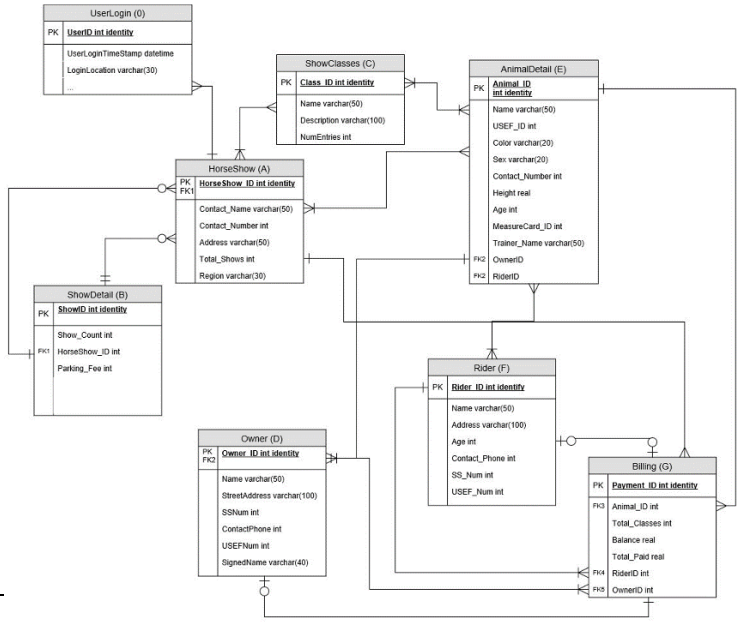
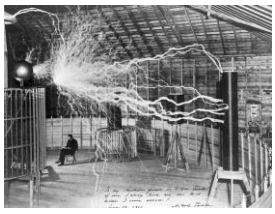
```
#=> EXAM Part 0 - fill in the blanks
     obj_Name   | charcter code      |   explicit code
      --------- | ------------------ | ---------------------------
  i) mytuple =  |  (, )              |=> mytuple = tuple(myobject)
 ii) mylist =   |                    |=> mylist  =
iii) mydict =   |                    |=> mydict  =
 iv) myset  =   |                    |=> myset   =
  v) dataframe =|                    |=> df      =
 vi) mystring = |                    |=> mystring=
# data files here in either a zip or individual files. if you download git
#https://github.com/bbe2/IT.304.Fall.2022/tree/Shakespeare-Corpus
#https://github.com/bbe2/IT.304.Fall.2022/blob/Shakespeare-Corpus/shakespeare_txt_fullname.zip
#-------------------------------------------------------------------------------
#=> EXAM Part I    - replicate shakespeare spreadsheet from wks 5-8
#-------------------------------------------------------------------------------
  #    Situation: unfortunately your business customer cant read the data b/c
  #    they dont know how to open text files in WORD. PLEASE help them out!
  #      1a) get the names of the play
  #      1b) create a list of play script data
  #      1c) count total words in script and titles
  #      1d) create numeric indexID for each: hint -> list(range(999))
  #      1f) DARN-it! there is not play type information. What should u do?
  #      1e) create a dictionary that matchs weeks 5-8 input spreadsheet
  #          => title, script, type, id
  #      1g) send dict to df, df to spreadsheet, email to me
#-------------------------------------------------------------------------------
#=> EXAM Part II   - create summary report by play type
#-------------------------------------------------------------------------------
  #       Total all script words and title words by 3 play types
  #       send to df to spreadsheet and email to me
#-------------------------------------------------------------------------------
#=> EXAM Part III  - ask user what play they want to read and email the data
#-------------------------------------------------------------------------------
#       3a) Create an object with one or two functions.
#           Ask user what play they want to read.
#           Figure out a minimum of 1 other useful piece of information
#           to display or include in user report.
#           Have function export data and send me data file.
```

Wk	Focus & Medium	Weekly Topic & Assignment
9 **Oct 24**	**Objectives**	**1)** Pillars of Python exam **(due 11/4 latest)** **Goal 1:** illustrate skill evidence of data pack and unpack **Goal 2:** create a transaction generator **class** object and 2-4 functions that asks the user a data question of your

to 29		choice. Then package the data, display it on screen, and send a report to a spreadsheet and a text file. Upon review of all code covered and your excellent progress, I decided to add one new zipper section with data pack and unpack options. The updates will help you with the exam and **ANY** time in the future with data pack/unpack. ~ ~ ~ ~ ~ ~ ~ We are transitioning back to our systems analysis and design models but will integrate our python learnings into ongoing exercises and class discussions.

Assignment: Readings		**2) Data Flow Diagramming**
		• Review Model.4.DFD handout (provided) <how.to.doc link> is next bullet!
		• Read: MIT System Design –ch6, p.1-12(in detail) skim till p.18
		• Re-read Tilley, ch5 p144 – 146, 152 – 163
		  ○ note the similarity on page 163 to youtube video
		• Watch: Systems Analysis and Design Ch10, j. barlow, 09.07.2016
		  ○ https://www.youtube.com/watch?v=ztlQvpS4QHk
		  ○ pay **extra** attention from min 10:31 till end of video
		  ○ note the use of relational database table diagraming
		~ ~ ~ ~ ~ ~ ~

In-class Discussion Questions		**Questions:** (in class discussion)
		1. What would be a primary way to generate Python transactions representing the education DFD at the end of the video (@ 10 min time point) and Tilley p.163?
		2. What kind of report would a school administrator want to see?
		  1. We will outline/draw together, so please have 1-2 ideas.
		3. What would you rather spend your time doing?
		  b. Making DFD's or designing data table relationships? (2nd example below)

relational database tables horseshow		
Nikola Telsa's Generator		

Wk	Focus & Medium	Weekly Topic & Assignment
8 Oct 17	(1 of 5) **Objectives**	**Congratulations!**

to 22

- dict = { }
- list = [ ]
- string = " "
- tuple = (,)
- set = set( )
- pd.DataFrame()
- series = d = {'a': 1, 'b': 2, 'c': 3}
  - ser = pd.Series(data=d, index=['a', 'b', 'c'])

What do you do if you need to view an object?

=> dir(myobject)

help(myanimal object)
<F9>

I am very proud of you, and have been incredible learning the "7 pillars of Python."

You have completed Python's core data objects, including

- reading any pd.read_csv('path') (comma separated values) or pd.read_xlsx() data from a spreadsheet application.
- understand positional indexing.
- learned axial data positioning with 0=row, 1=column.
- understand how to pack, unpack, and read Python-packed data objects.

```
#===================================
            [({...})],{"" : [ ] }
#===================================
          string data is in a dictionary {key:value}
            which is inside a tuple
             which is inside a list
              separated by a comma to another object
               which is a dictionary with
                a string for a key, and
                  a list of its key values
```

Now you have the tools to decipher how data is packed and figure out how to mix and mingle python objects and re-organize as needed.

You have also worked with iterators, conditionals, and variables and can transpose data

```
==========================================
```

dir(<myObject>) => displays its constructors, methods, and attributes

```
['__class__', '__delattr__', '__dict__','__dir__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__',
 '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__',
 '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__',
 '__repr__','__setattr__', '__sizeof__','__str__', '__subclasshook__',
 '__weakref__',
 'name',
 'species',====> user defined attributes
 'train']
```

```
help(<myobject> or <function)
-------
 |  Data and other attributes defined here:
 |  name = ''
 |  species = '' ==> these are the attributes in our wk7 object
 |  train = ''
```

Wk	Focus & Medium	Weekly Topic & Assignment
8 Oct	(2 of 5) [wk8 code git]	""" it.304.wk8 (10/16-10/22/22) Created on Sat Oct 15 13:56:24 2022

17 to 22		@author: 17574 b.hogan@snhu.edu
		"""
		#======================== > Week 8
		#===== Classes - Week 8
		#================
		#=======================
		
		#=> Objective: use the following Classes example to make one of your own
		#=> new function input("<message>") -> asks user for a value
		
		#===
		# Part I: Import libraries and source data
		# Part II: Draft an object with couple functions
		# Part III: Creat a child object and run the function
		# Part IV: Run a report
		#===
		''' CARLY! this is not boo scary!
		conditionals (below)are a set of questions, often in your own words.
		if you are stuck, set a timer and spend no more than 20 minutes.
		research says your better phoning or emailing a friend as anything
		after 20 minutes exceeds optimal learning. good luck!
		
		#=> # Part I:
		#==
		#=> # Part I: Import libraries and source data
		#==
		Import libraries + data
		import pandas as pd #dataframe library
		import numpy as np #numeric library
		import matplotlib.pyplot as plt #visualization library
		import os #operating system library
		import sys # sys.exit()
		os.chdir('c:\\Users\\17574\\Desktop\\data_it304') #microsoft uses 2\\
		df0 = pd.DataFrame() #explicitly set datafarme
		df0 = pd.read_excel("data_shakes_corpus_v1.xlsx") #ETL method 2
		df0.info()# RangeIndex: 37 entries,0 to 36, 4 col=> all data u need to index
		mydict = df0.to_dict() #df to dict
		'''mydict_shakespeare => {'title':{},'script':{},'type':{},'ID:{} }'''
		len(df0) #37 lenth is always veritical by default!
		
		#=> # PART - DEtour - was best to add NEW INFO here
		'''this will help you create a report for quiz end of wk...'''
		#================================>
		#=> Function idea and drive a bitchen camero data to excel
		#================================
		# Use if, elif, else 'conditionals' to draft your questions based on data
		# Consider drafting 1-3 questions on an index card before coding
		# detail what information need to perform so you focus vs get stuck on names
		# remember - objects are the actors and functions are their script
		
		'''Fucntion ideas & examples:
		i) write a function to count total characters in a play or all plays!
		ii) use an iterator, count characters, and put in a list
		iii) use new lists to create a report or write back to excel using'''
		
		mylist = [] #so this could be a function to count characters
		for i in mydict['title'].values(): mylist.append(len(i))
		print(mylist, type(mylist), sum(mylist))

8 Oct	(3 of 5) wk8 code git	

17 to 22

```
#use the new objects and variables to creat a dictionary
myNewDict = {sum(mylist): mylist}
        #   or {'play-1':[<TitleTotalWords>,<ScriptTotalWords>]}

print(myNewDict)
print(type(myNewDict))
myDF = pd.DataFrame.from_dict(myNewDict)#function create a pandas.DF from
dict
#myDF.info()                                #check it out

''' Send to excel or view here - will review in class'''
mywriter = pd.ExcelWriter('myoutput.xlsx') #create object that writes out
myDF.to_excel(mywriter)
mywriter.save()
myDF                                    #Excel will look exact same !

#=> # Part II:
#============================================================
#=> # Part II:  Draft an object with couple functions
    # We are training with .self notation. write self.<attribute or
variable>
    # are inherent, or part of our instantiated children objects
#============================================================

'''START - HIGHlight all of class and hit F9 from lines 93 to 150 '''

class shakespeare_minion:           #this defines the parent object
    pass
    name = ""
    perform_work = 0                #yes,no switch so could exit terminal
    total_plays_not_read = len(df0) #use an object vs. hardcode a value
    total_plays_read = 0            #increment so you know how much work
done
    num_plays_work_now = 0          # countdown tracker based on user input

    '''Function-1: ask user how many plays to read'''
    def how_much_work_master(self):
                        #int() function here ensures user response encoded as
a #
        perform_work = int(input("Enter greater than 0 to run program =>
"))
        if perform_work <= 0:
            sys.exit()       #On/off switch so can exit program in terminal

        if perform_work > 0:    #NEW - ask user a question with input()
            self.num_plays_work_now = \
                int(input("Enter how many plays you will read today?=> "))
        perform_work = 0    #set back to zero as 1x trigger









    '''Function-2: have minions completed what they said they would do?'''
    def do_work_and_report_status(self):

    #0) for transactions, here would be some kind of wait time to do work


    #1) condition 1 - Did we complete total work yet?
```

8 (4 of 5)

Oct 17 to 22	wk8 code git	

```
        if self.num_plays_work_now <= 0:
            #after test, then increment/decrement associated variables
            self.total_plays_not_read = self.total_plays_not_read - 1
            self.num_plays_work_now = self.num_plays_work_now - 1
            total_plays_read =+1 #another way to increment variables

            return "Master! {} is done. I finished {} plays today.". \
                            format(self.name,self.total_plays_read)

    #2) condition 2 - Still doing daily work ?
        elif self.num_plays_work_now > 0:
            #after test, then increment/decrement associated variables
            self.total_plays_not_read = self.total_plays_not_read - 1
            self.total_plays_read = self.total_plays_read +1
            self.num_plays_work_now = self.num_plays_work_now -1
            total_plays_read =+1        #another way to increment variables

    #3) condition 3 - this is a NESTED loop b/c now you either no more work
    #       or you report what you have left to do in this batch
            if self.num_plays_work_now == 0:
                return "Master I have {} plays left to read AND no more
work.\
                            I am 100% done for today so start over!".\
                                        format(self.total_plays_not_read)

            else:
                return "Master I read {} plays today and have {} more plays
\
                to do in this most egregiousness and unjust batch.".\
                    format(self.total_plays_read,self.num_plays_work_now)


'''END HERE - HIGHlight all of class to define full object'''

#============================================================
# Part III: Creat a child object and run the function
#============================================================

# IIIa: ask user number plays to ready & run the transaction
'''Run these 3 lines together! - This starts to queue up total work'''
minion = shakespeare_minion()
minion.name = "Toothless Harold"
minion.how_much_work_master()         #ask user how much to do!

#=================
'''====>Now run a transaction, that is read a play.
        this program runs these transactions manually.
        The final little program we make will run them all at once.'''

#================= select all 4 lines - keep running to run out of work!
print(minion.do_work_and_report_status())
print(minion.total_plays_not_read)
print(minion.num_plays_work_now)
print(minion.total_plays_read)
```

Wk	Focus & Medium	Weekly Topic & Assignment

8

Oct 17 to 22

(5 of 5)

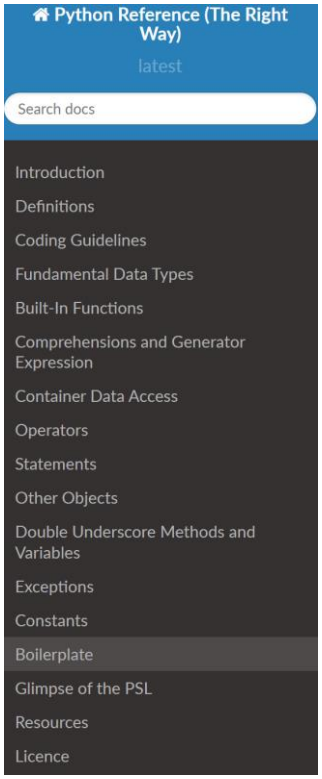
Blog discussion of pros \ cons searching for information to help learn clases

cliff-notes, even this 'ok' site is there to sell and grab your computer data. Always use a vpn!

Sometimes you need a quick answer.

But – when you start something new spend time on your cheat sheet and find quality resources so in times of need you can make it happen.

[mit student gem]



b.hogan 11:06 AM

autobots.304 - I am not against the internet for training and checking things out but everyone remember what Jackson said in class, "often I can get distracted with other things."

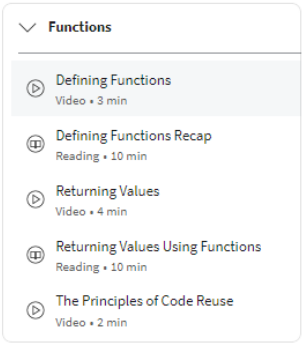
This is a significant challenge for your generation and learning what is good, bad, and ugly infomation is usually usually easy for 'ugly' information and questionable for all else. For instance, this article is decent for what we are doing for week 8 as you get busy with classes. The examples are informative, complete, and meaningful. I would be comfortable putting as a syllabus reference BUT it isn't a quality academic reference.
https://www.toptal.com/python/python-class-attributes-an-overly-thorough-guide (edited)

> **Toptal Engineering Blog**
> Python Class Attributes: An Overly Thorough Guide
> Python class attributes can lead to elegant code, as well as frustrating bugs. In this guide, we will outline specific use-cases for attributes, properties, variables, objects and more. (87 kB)

A MUCH better way is to go onto cousera and find a class that is similar to what you are learning whne you need to, ah - do things more quickly lets say.

So it took me 2 minutes to find on Coursera "crash course in python." Here is what they ahve in week 2 - anything look familar to the zipper?

image.png

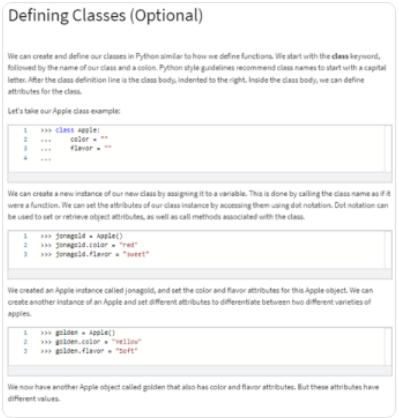


Sure, its costs $39 a month to have access but for Week 8 you are now writing your own functions while learning your own python objects. For our class purposes this is nice supplemental information but I am also giving you evcerything you need, or at least I hop so, to be productive.

As promised, we will review such deft research approaches in November. I keep hitting this nail because I don't want you to experience what I did, that is not going outside, while working harder and not smarter for parts of my coding and analysis work life.

for this week - here is what the google coursera class has for class attributes like we reviewed yesterday for our farm animal names and species. It is nothing special, and I like my better.

image.png



Thank you for reading! my purpose is to keep hammering this so you spend more time performing quality work

example ok website https://www.toptal.com/python/python-class-attributes-an-overly-thorough-guide

Wk	Focus & Medium	Weekly Topic & Assignment

7
Oct 10 to 15
	(1 of 4)
	
	QUIZ Instructions
	QUIZ Answer
	
	Objective: more exercises on python pillars to prepare for creating an object generator.
	
	We will review in class but you will need to answer and turn it in when finished. Turn it in by the 19th the latest but won't take you long.
	
	• I will post everyone's own gradebook this week.
	
	• The 2nd part of the week will review class objects
	
	 
	versus
	 
		"""# -*- coding: utf-8 -*-
		Created on Mon Oct 10 10:59:53 2022
		@author: 17574
		==========
		#==========
		#==
		#=> it.304 2nd Graded Assignment
		#==
		#=========="""
		import pandas as pd #dataframe library
		import numpy as np #numeric library
		import matplotlib.pyplot as plt #visualization library
		import os
		os.chdir('c:\\Users\\17574\\Desktop\\data') #microsoft uses 2 \\
		df0 = pd.DataFrame() #explicitly set the data object
		df0 = pd.read_excel("shakes_corpus_v1.xlsx") #ETL method 2
		df0.info()
		mydict = df0.to_dict()
		#==================================
		#=>1.0 Pillar: Iterators
		'''1.1 Task: use an iterator and produce total words all plays'''
		#==================================
		
		#==> ENTER YOUR CODE HERE
		mylist = []
		for i in mydict['script'].values():
		  mylist.append(i)
		total_script_characters= 0 #how many characters?
		for i in mylist:
		  total_script_characters = total_script_characters + len(i)
		total_script_characters
		
		# Answer: 1,212,379
		
		'''1.2 Task: what is easiest in code to double total characters'''
		#==> ENTER YOUR CODE HERE
		
		total_script_characters*2
		
		# Answer: 2424758
		
		#==================================
		#=> 2.0 Pillar: Functions
		'''Task: Generate a tuple wth the code provided
		  hint: use codebook '''
		#==================================
		mylist = []
		mytuple = ()
		for i in range(37):
		  mylist.append(i)
		
		#==> ENTER YOUR CODE HERE
		mytuple = tuple(mylist)
		
		# Answer:
		print(mytuple) # (0,1,.....36)
		print(type(mytuple)) # tuple

7

Oct 10 to 15

(2 of 4)

Hacksaurus

versus

missing Danny and Jackson memes

```
#====================================
#=> 3.0 Pillar: Built-in objects - Sets
#====================================
''' 3.1 Quickly explain what this statement is doing

    random.randint(len(mydict),len(df0['script']))

    3.2 What does the type() function tell you and why is it
        important?

    3.3 Create one set from =mydata1 and mydata2
    3.2 Use the type() function to prove it is a set
    3.5 Why is performing housekeeping a good habit?'''
#==================================

import random  # generates random numbers
               # randint(start value, end value)
mydata1 = random.randint(len(mydict),len(df0['script']))
print(len(mydict),len(df0['script'])) #4, 37

#==> 3.1 ENTER YOUR RESPONSE HERE
'''pulling random value from 4 to 37'''

#==> 3.2 ENTER YOUR RESPONSE here after the 3 lines of code
type(mydata1)
mydata1 = (mydata1,)
type(mydata1)

'''can only add objects that are the same object type'''

#==> 3.2 ENTER YOUR RESPONSE HERE
mydata2 = 1,2,3,4,3,2,1
myset = set(mydata1 + mydata2)

#...ANSWERS:
#Answer: <your code answers should be the same except m
       #each person will have 1 diff value
print(mydata1,set(mydata2))     # 35, {1,2,3,4}
print(myset)                    # {1, 2, 3, 35, 4}
print(len(myset))   # 3.1 => 4
print(type(myset))  # <class 'set'>

#Answer built in objects only take one parameter.
# BUT you can add objects together as long as they are the same
# object type.

# housekeeping
#Why: so dont absob data you dont need later by accident
del mydata1; del mydata2;del myset

#==================================
#= 4.0 Pillar - interpreting packed built-in objects
'''Task: you have the following object visible to your in your
   'variable explorer' window. if script is in the ... describe
   the object container around it and what you would do to
    unpack it.'''
#==================================
'''             [({...})],   '''
#==================================
the string data is in a dictionary
which is inside a tuple
```

		which is inside a list

Wk	Focus & Medium	Weekly Topic & Assignment

7
Oct 10 to 15
	3 of 4)
	
	classes!
	
	this is is not using the self parameter so functions are outside of the object
		"""# -*- coding: utf-8 -*-Created on Mon Oct 12 10:59:53 2022
		@author:17574 b.hogan@snhu.edu it.304.fall.22
		# WEEK 7 CODE final - including classes """
		#===
		#=>week 7 Object Classes Overview
		#==================================
		
		Lexical Analysis
		always remember about indent \ dedent!
		if you copy and paste and teh spacing is wrong it wont run
		
		https://python.readthedocs.io/en/latest/reference/lexical_analysis.html
		
		#Create a report structure
		mydict = {"training done":[], "total animals":0}
		class myFarm: #create parent class object
		pass
		name = ""
		species = ""
		train = ""
		def add_train(traintype): #create a user function to count, sort
		mydict["training done"].append(traintype)
		mydict["total animals"] =+1
		
		#-------------> #children instantiate from parents
		a1 = myFarm() # instantiate children objects from parent, a for animal
		a2 = myFarm() # all object names are user defined
		
		#update attributes
		a1.name = 'mackenzie' #object.attribute or object.function
		a1.species ='dog'
		a1.train = 'speak'
		add_train(a1.train) #cheCK-OUT! <only here bc space>
		
		a2.name = 'vinny'
		a2.species = 'horse'
		a2.train = 'jumping'
		
		add_train(a2.train) #'''function accepts attribute to update dictionary object'''
		
		#write a simple report using a dictionary data object format
		mydict_rpt = {a1.name:a1.species, a2.name:a2.species,"metrics=>":mydict}
		mydict_rpt
		'''{'arnold': 'dog','vinny': 'horse','metrics=>':
		{'training done': ['catch', 'jumping'], 'total animals': 1}}'''
		
		#use object's constructors to view its contents
		print(a1.__dict__,a2.__dict__)
		''' {'name': 'arnold', 'species': 'dog', 'train': 'catch'}
		{'name': 'vinny', 'species': 'horse', 'train': 'jumping'}'''
		
		dir(a1)
		['__class__', '__delattr__', '__dict__','__dir__', '__doc__',
		'__eq__', '__format__', '__ge__', '__getattribute__', '__gt__',
		'__hash__', '__init__', '__init_subclass__', '__le__', '__lt__',
		'__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__',
		'__repr__','__setattr__', '__sizeof__','__str__', '__subclasshook__',

7	Oct 10 to 15	'__weakref__', 'name', 'species', 'train']

```
#============================================================
                        #=>Week 7 Objects part II
        #============================================================

#==> this is using programmming construct of .self.

class dog_train:
    name = ""
    num_fetch_train = 30
    num_fetched = num_fetch_train
    trainer_ok = 0

    def fetch_train(self, num_balls):
        self.num_fetched = self.num_fetched - num_balls
        if self.trainer_ok == 0 and self.num_fetched <= 0:
            return "sorry! {} not fetch trained. {} balls over a target of
{}".format(self.name,abs(self.num_fetched),self.num_fetch_train)
        elif self.trainer_ok == 1:
            return "Whew! {} passes training after {} balls".format(self.name, abs(self.num_fetch_train-
self.num_fetched-1))
        else:
            return"{} on target to pass fetch train with {} balls
left".format(self.name,self.num_fetch_train-self.num_fetched)

dog1 = dog_train()
dog1.name = "cheeseman"
print(dog1.fetch_train(9))
print(dog1.fetch_train(31))
dog1.trainer_ok = 1
print(dog1.fetch_train(1))
==============================
```

Class, object, and function definitions:

Classes – are a framework or template for creating objects, attributes, and methods.

Objects – are the actors performing work. Child objects instantiate from parent objects and may contain their attributes and methods or have distinct attributes and methods.

Methods - are object instructions detailing how to perform behaviors in a class such as data arrangement, computation, printing, and conditional logic trees, perhaps to test, parse, or look for specific information. Methods do not have to return a value!

Functions – a set of instructions to accomplish a task independent of an object and typically part of a program. They may accept arguments and always return a value.

Class attributes – user-defined names that describe features of a class, and methods can use their values. For example, an object's unique ID, color, name, or numeric value for use in a calculation.

.self <self.attribute> is the first argument in a class function identifying its own attributes.

Essential Python tools associated with objects.,

Built-in types – Python core boolean, comparators, numeric types, and operations like 1+1, iterator types, and operations. REVIEW recommended!

Python Essential Data structures – lists, tuples, sets, dictionary, looping, more on conditionals. Methods and tips and tricks.

Wk	Focus & Medium	Weekly Topic & Assignment

6
Oct
3
to
8
	(1 of 4)
	
	git codebook
	
	wk6.d2.lecture
	python pillars
	family
	
	10/5 Class
	-> do all the basic Python pillars
	• **core objects**
	• **conditionals**
	• **iterators**
	• **functions**
	• **transposition**
	
	10/6 Class
	-> ~~create objects and functions for reporting~~
	> all pillars except transformers and classes
	
	10/8/22 – wrap-up
	
	• We completed code on left for week6.
	• This sets us up for making transaction generator and finally advaning our work to system design were we will pull project plans, manipulate them, and mock up reporting.
	
	]
		"""# -*- coding: utf-8 -*- Created Sep 15 07:58:23 2022
		@author:17574 b.hogan@snhu.edu it.304.fall.22
		Objective: import data and apply zipper to transform, iterate,
		use conditionals, apply functions, leading to python classes work
		Library homebase = Python package index: https://pypi.org
		"""
		'''====================
		#=========================
		#==
		#=>STEP 1 get pip library install path from
		#==
		#=========================
		#===================='''
		import pandas as pd #dataframe library
		import numpy as np #numeric library
		import matplotlib.pyplot as plt #visualization library
		import os
		os.getcwd() #where am i? <get working directory>
		#os.chdir('c:\\Users\BBE\DATA\') #some op.sys use one slash
		os.chdir('c:\\Users\\17574\\Desktop\\data') #microsoft uses 2 \\
		os.getcwd()
		
		df0 = pd.DataFrame() #explicitly set the data object
		#df0 = pd.read_csv("shakes_corpus_v1.csv") #ETL method 1
		df0 = pd.read_excel("shakes_corpus_v1.xlsx") #ETL method 2
		df0.info()
		# RangeIndex: 37 entries, 0 to 36
		# Data columns (total 3 columns):
		# # Column Non-Null Count Dtype
		# --- ------ -------------- -----
		# 0 title 37 non-null object
		# 1 script 37 non-null object
		# 2 type 37 non-null object
		# 3 ID 37 non-null int64
		# dtypes: int64(1), object(3) memory usage: 1.3+ KB
		
		print(type(df0)) #use type() to always see what an object is
		df0.head()
		# title ... type
		# 0 Alls Well That Ends Well ... Comedy
		# 1 As You Like It ... Comedy
		
		#2.1 use pandas df.to_dict() to move data into dictionary object
		mydict = df0.to_dict()
		print(mydict.keys()) #['title', 'script', 'type', 'ID'])
		type(mydict.keys()) # object itself is keys
		
		#2.2 understand what a dictionary and zip is doing
		mylist_keys = list(zip(mydict.keys()))
		mylist_keys # [('title',), ('script',), ('type',), ('ID',)]
		
		#Inspect huge data and then break into smaller chunks
		mylist_values = list(zip(mydict.values())) #WOW huge !
		#point - zip helpful but continue to learn more functions
		
		mylist_values #=======================> #MEGASAURUS
		# 35: 'Tragedy',

wk 6 Oct 3 to 8

]
]]
][][]][p[p’’

NOTES ARE BELOW TO HELP YOU WITH YOUR HOMEWORK

PLEASE BE CREATIVE

week 6
(2 of 4)

PLEASE BE CREATIVE

```
                    # 36: 'Tragedy'},),
                    #{0: 1,
                    # 1: 2,
                    # 2: 3,
'''====================
#=========================
#============================================================
#=>STEP 2 - seperate Megasaurus into usuable object chunks
#============================================================
#=========================
#===================='''

'''2.1'''
type(mylist_values) #=> [({...})],

'''======> packed as [({...})], =>list, tuple, dictionary'''

type(type(mylist_values[1]) )#hmm doesn't unpack
len(mylist_values)  #=> 4 columns in spreadsheet, ie data objects

'''megasaurus - all plays and words'''
mylist_values
                        # => format is list[(tuple(dict))]
                        # [  ({id:title}),({id:script}),
                        #    ({id:type}), ({id:id}) ]
                        # zip added an key sequential value
'''==>2.2'''
'''use slicing [0:1], [2] to view next level down'''
type(mylist_values[0])  # tuple
mylist_values[0]        #=> [x] is called slicing

                  Out[23]:
                  ({ 0: 'Alls Well That Ends Well',
                     1: 'As You Like It',

'''now think data like in spreadsheet'''
#  columns
#    0      1     2      3
# |title |script| type |  id  |
#  hamlet,oh joy,tragedy, 29

mylist_values[1] #displays all the script text!

'''==>2.3'''
len(mylist_values[1])  # waits its '1' so need to unpack my data

mylist = []
for i in mydict['title'].values():
    mylist.append(i)
mylist
len(mylist)  #37 - does htat match spreadsheet? always know your bounds

title_total_characters = 0    #how many characters?
for i in mylist:
    title_total_characters = title_total_characters + len(i)
title_total_characters  #do you get 560 ?

'''============================================================
==>2.4 autoBOTs304 - repeat this for total script words
#============================================================
===> moved  this into the graded_assign_wk7'''

#====================
#=========================
```

wk
6
Oct
3
to
8

week 6
(3 of 4)

CREATIVITY ENCOURAGED

[wk6_shakespeare codebook]

```
#=========================================================
#=>STOP! : view 'Variable Explorer' window
#  use this feature to propel data transformation learning
#=========================================================
#=========================
#====================

'''#====================
#=> WRAP - UP  Housekeeping
#  delete variables not using; help avoid unnecssary mistakes
#==========
# be mindful how you stage both variable and data names
#   df0 = baseline import
#       df1 = analysis 1
#           df2 = analysis 2
#==================='''

'''==>2.5'''
del mylist_keys    # del removes a variable

'''
mylist2 = []
for i in [mydict.get('title')]:
    mylist.append(i)   #so what happended here a. wrote name list wrong
print(len(mylist2), len(mylist))
#make a note here on what happended..............
mylist    #stacked a list on a dictionary bc meant to use list2
'''
#go back and rest data for part 2
mylist = []
for i in mydict['title'].values():
    mylist.append(i)

'''====================
#=========================
#=========================================================
#=>STEP 3:  Use dir(object) to learn its methods to get work done
#=========================================================
#=========================
#====================='''
'''==>3.1'''
#===============> use dir() to get functions available for an object
myset = set()
print(type(myset))
dir(myset)
 # '__xor__',  ==> these are constructors, more later
 #    'add', 'clear',    ==> these are methods
 #   'copy','difference', 'difference_update', 'discard',
 #  'intersection','intersection_update', 'isdisjoint', 'issubset',
 #   'issuperset','pop', 'remove','symmetric_difference',
 #   'symmetric_difference_update','union',  'update']'''

'''==>3.1'''# ====> SETS
mylist2 = mylist
mylist2.append("Winters Tale")  #add one duplicate title
myset = set(mylist2)
print(len(mylist),len(myset))  #so got rid of duplicate
del mylist2

#============> ACTION learn what you need and go find it
mystring = ""
print(type(mystring))
dir(mystring)
#'''__subclasshook__',  'capitalize', 'casefold',, 'center',
```

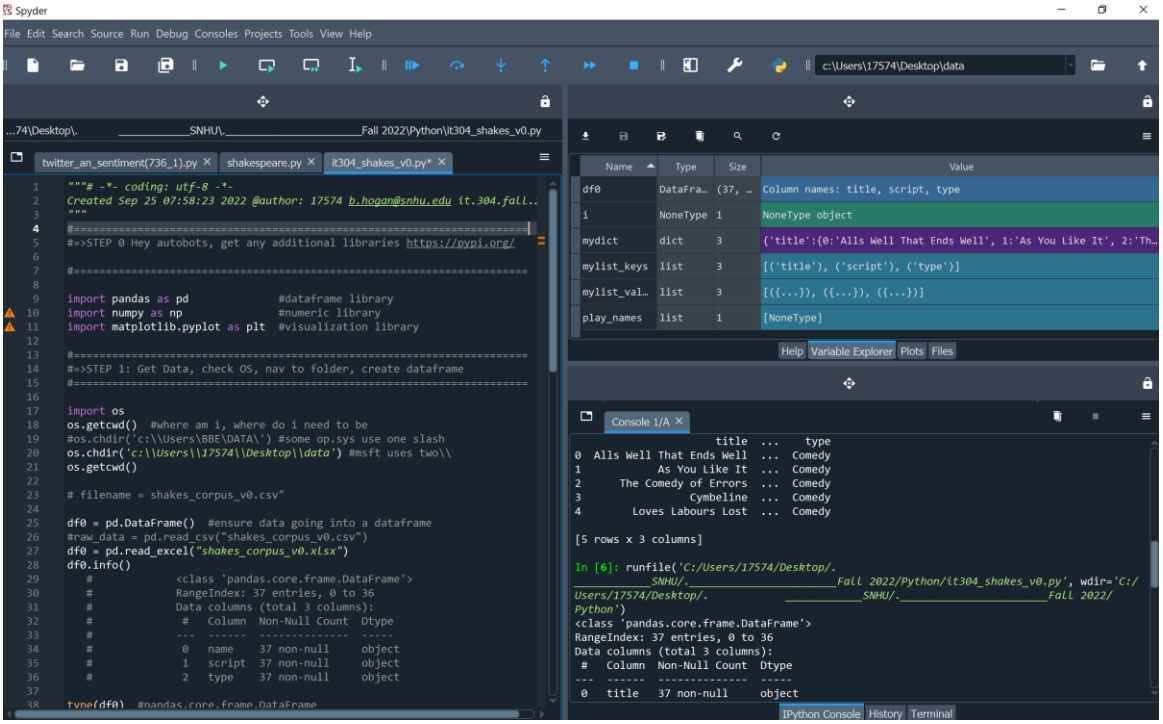
wk 6 Oct 3 to 8

week 6
(4 of 4)

CREATIVITY ENCOURAGED

```
#'count', 'encode', 'endswith', 'expandtabs', 'find', 'format',
#'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal',
#'isdigit', isidentifier', 'islower', 'isnumeric', 'isprintable',
#'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip',
#'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust',
#'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines','startswith',
#'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']'''
'''====================
#=========================
#============================================================
#=>STEP 4: More dictionary: .keys(), .values(), .get(<key>)
#============================================================
#=========================
#===================='''

'''==>4.1'''
mydict.get('title')  #.get() views one series
play_names = [mydict.get('title')]
play_names
        [{0: 'Alls Well That Ends Well',
          1: 'As You Like It',
          2: 'The Comedy of Errors',
mylist
# Now add titles to a different object with an iterator
mylist2 = []
for i in [mydict.get('title')]: #method returns a dict obj
    mylist2.append(i)
mylist2
    [{0: 'Alls Well That Ends Well',
      1: 'As You Like It',
      2: 'The Comedy of Errors',

#3.2 =>  Learn dictionary key, value, items parameters
mylist_key =  []
mylist_values = []
for k,v in mydict.items():
    mylist_key.append(k)
    mylist_values.append(v)
mylist_key              #['title', 'script', 'type', 'ID']
mylist_values   #'''again megasaurus'''

'''==>4.2''' #=>  Understand and count items in a list
len(mylist_values) #hmm  why is this only four ?
mylist_values[0]
mylist_values[1]
mylist_values[2]
mylist_values[3]

#====================
#=========================
#============================================================
#=>STEP 3:  Use Functions and get Meta Data
#============================================================
#=========================
#====================
https://docs.python.org/3/library/functions.html#built-in-functions

sum(mylist_values[3])-1
sum(df0['ID'])-1
len(set(df0['ID']))
```

==========> END week 6

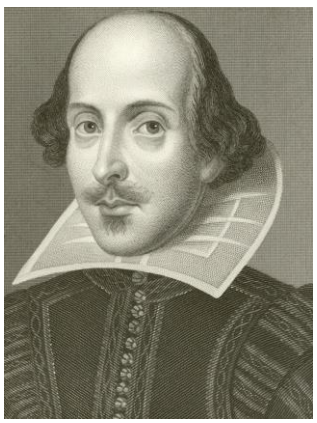
Wk	Focus & Medium	Weekly Topic & Assignment
5 **9-26-10-1**	(1 of 3) 9/28 Class – Jupyter lite not working. **Spyder IDE** going forward polyglot = knowing or using several languages: Everyone set their window up this way **Sypder Basics**	**Objective =** entire class get on Sypder IDE for consistent training **A: Install Sypder:** https://docs.spyder-ide.org/current/installation.html • PyCharm is an IDE for polyglot programming,ie > 14 languages. • As such we will use Spyder IDE, great science + students **B: setup Spyder windows like this** **Left** => code; **top right** =>variable explorer; **lower right**=>console **C: install packages:** always run library imports first. If one doesn't run then go to termind on this screen and pip install from https://pypi.org/,  **D: Sypder interface basics** **a)** code window opens any .py file with code assist editor **b)** highlight code you want to run and hit function F9 **c)** in the console you see the output! that simple **d)** Variable explorer NEAT bc tracks all the objects and current status of a variable

		d.1) you can click on a variable and it opens a window so you can see the contents. **e)** I appreciate we discussed need to code without applications but this application serves to reduce the basic visual output burden of the code you write. You still need to write the code to create an manipulate the data objects which is the core skill.

Wk	Focus & Medium	Weekly Topic & Assignment

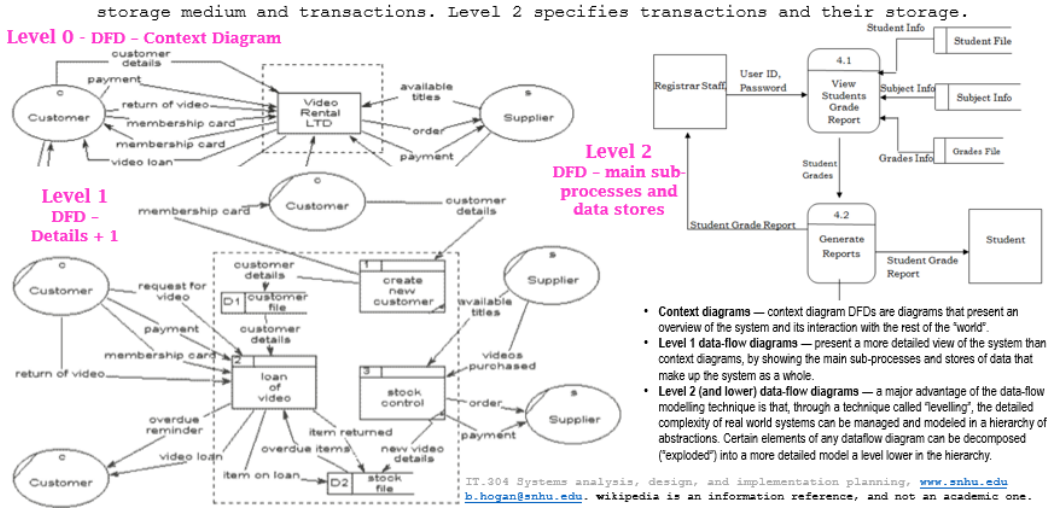
5
 	
9-26 - 10-1
	(2 of 3)
	Shakespeare Corpus
	Class Team Coding
	09-28-022
	
	Step 1: libraries
	`#dataframe library`
	`import pandas as pd`
	
	`#numeric library`
	`import numpy as np`
	
	`#visualization library`
	`import matplotlib.pyplot as plt`
	
	`#operating system`
	`import os`
	
	➢ Reading the data
	➢ Use conditional to loop words
	➢ Make fun graph
	➢ transpose data between lists, dictionary, string, tuple
	
	OUT[10]: tip!
		Objective = begin working with 5 pillars of python; create data folder on c:\drive. Code -> Interpret ->
		
		 
		
		Step 1: change directory, get corpus file path
		`import os  #operating system library`
		`os.getcwd()  #command to get workiing directory`
		
		q1> What do bad characters in your paths do? A: cant read data
		`In [2]: runfile('C:/Users/17574/Desktop/.___SNHU/.___Fall 2022/Python/it304_shakes_v0.py', wdir='C:/Users/17574/Desktop/.___SNHU/.___Fall 2022/Python')`
		`File "<unknown>", line 23`
		`SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape`
		`os.chdir('c:\\Users\\17574\\Desktop\\data') #msft uses two\\`
		`os.getcwd()`
		`df0 = pd.DataFrame()  #ensure data going into a dataframe`
		`#raw_data = pd.read_csv("shakes_corpus_v0.csv") #oops doesn't work`
		`df0 = pd.read_excel("shakes_corpus_v0.xlsx")  #this works!`
		`df0.info()`
		`<class 'pandas.core.frame.DataFrame'>`
		`RangeIndex: 37 entries, 0 to 36`
		`Data columns (total 3 columns):`
		` #  Column  Non-Null Count  Dtype`
		`---  ------  --------------  -----`
		` 0  name  37 non-null  object`
		` 1  script  37 non-null  object`
		` 2  type  37 non-null  object`
		`memory usage: 1016.0+ bytes`
		`type(df0)  #pandas.core.frame.DataFrame`
		`df0.head(2)`
		`name  ...  type`
		`0  Alls Well That Ends Well  ...  Comedy`
		`1  As You Like It  ...  Comedy`
		
		q2> What happens when you dont have a cheatsheet and need to convert a dictionary to a list? Python Convert Dictionary To List - Python Guides A: === ACTION = email brian this answer
		<==========ACTION
		`mydict = df0.to_dict()`
		`print(mydict.keys())`
		`out[10]: dict_keys(['title', 'script', 'type'])`
		
		`mylist_keys = list(zip(mydict.keys())) #hmm my data columns looks good`
		`mylist_keys`
		`OUT[10]: [('name',), ('script',), ('type',)]`
		`#DANGER Will Robinson this is a megasaurus`
		`mylist_values = list(zip(mydict.values())) #holy cow this is huge!`
		`mylist_values=====> this is huge, make sure you undertand`
		
		`#finally break data into more manageable things to do`

	going forward will use python [out] to signify output
	
	#now as a class we will experiment with cheatsheet
		mydict.get('title') #learn a new function
		play_names = [mydict.get('title')]
		play_names OUT[10]: [{0: 'Alls Well That Ends Well',
		1: 'As You Like It',
		2: 'The Comedy of Errors',
		
		for i in play_names:
		print(i)
		Out[27]: [{0: 'Alls Well That Ends Well',
		1: 'As You Like It',

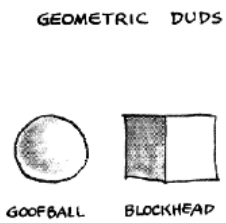
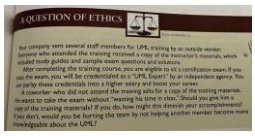
Wk	Focus & Medium	Weekly Topic & Assignment
5 9/26 - 10/1	**Focus Overview** **Python 101 coding**  shakespeare	**Goal:** build competence with Python built-in objects to manipulate data like working in a spreadsheet application. Why? Spreadsheets are 3rd tier objects versus primary information formats like databases and data objects such as Python's list, string, tuple, dictionary, and sets. And pandas series and dataframe objects. Why Python? ○ Its versatile, used across industries, and provides easy to learn data ETL (extract-translate-load), analysis, and reporting. ○ Manipulating data in objects make you more agile and confident grab.get data from anywhere. ○ Developing transposition skills with Python's data objects gives you the basic means to always work with any data in the future ○ These tools will help you perform system design and analysis with agility and deftness. ○ This is your new **HAMMER.** Now let's go **frame it.** The remainder of the course will use the following toolkit to perform system analysis & design exercises. **~~ Course System Design & Analysis Tooling ~~** **System Planning & Design** a) customer requirements outline with level 1 system diagramming methods (IDEF0, swimlanes, SWOT, etc) b) architect a system data flow diagram (DFD) b.1) key transactions b.2) key storage tables **Class Python Codebook** 1) data objects (list,string..) 2) user defined objects 3) iterators 4) conditionals 5) functions / methods 6) transposition 7) pandas dataframes\series 8) ETL **Preparation for our Shakespeare Assessment (given 9/30)** As discussed in class, you will be applying your learnings to the Shakespeare corpus by importing the data, performing transformations, and using iterations and conditionals to report on # characters, words, and # plays.

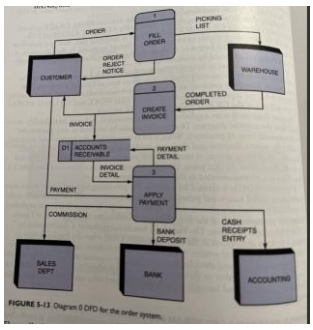
	corpus (git) **Assignment**	**Tasks:** • The zipper codebook has been updated <09.24.22> • Please work through the code examples again for 9/21 class • New: repeat 1 page of ETL, object, report code tasks ○ (due 9/30,posting shortly)

Wk	Focus & Medium	Weekly Topic & Assignment
4 9/19 - 9/24	**Overview** **Python 101 coding** **wk4 Assignment** Model.4.DFD Data Flow Diagram	Orientation to core Python functionality the course will use for system analysis and design projects. The codebook details core data objects, functions, iterators, conditionals, dataframes, and ETL. In short, everything you need to be successful in class and as an entry-level IT professional. Your objective is to "re-type" the code and bring class your learnings and questions for any code you do not understand. You are not learning code from scratch, but you need to understand and intuit the mechanics of iterators, if.elif.else conditions, and functions to perform work computational work effectively. I am 99.9% confident everyone can complete this work, and I hope everyone will have fun doing so. Good writing is good thinking, and good programming helps make IT work more meaningful and enjoyable. The latest version of the codebook, called the zipper, is in the bh.github. Enjoy the printed codebook handouts but ensure to update and print another copy in the upcoming weeks. The latest copy is always on the class git. Thank you for thoughtfully working through all codebook examples. Think about what the code is doing inside the computer. Write down anything that doesn't make sense for class discussion. Over the next few weeks we will learn the 7 pillars of python to build your representation of an data flow diagram.

		Model.4: Data Flow Diagraming <sparx-models> <website> <how.to.doc [VG]> <how.to.video> <wikipedia> Purpose: proess of representing simplified data transactions to help process and stakeholder owners agree on scope and boundaries of a systems analysis and design reengineering effort. Level 0 is the highlevel context. Key tasks are detailed in level 1 indicating storage medium and transactions. Level 2 specifies transactions and their storage.  IT.304 Systems analysis, design, and implementation planning, www.snhu.edu b.hogan@snhu.edu. wikipedia is an information reference, and not an academic one.

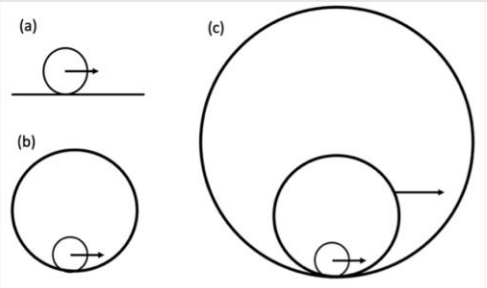
Wk	Focus & Medium	Weekly Topic & Assignment
3 **9/12** **-** **9/17**	**Focus** **Overview** **Reading Tilley, Ch6 entire chapter**	• Perform hands-on activities in Python to learn object-oriented programming(OOP) working with strings, dictionary, tuple, list, set, function, and objects. • As a team, outline system and code objects to simulate system analysis exercises. • Code is provided for you to re-type and learn. • Use cases will grow your confidence. Tilley details old and new techniques for systems modeling, like business process modeling (BPM) (ch1-2), data flow diagrams (DFD) (ch4), and data and process modeling (ch5). Exercises focus on techniques but with little substantiated in the field outcomes. Python hands-on OOP work will replicate varying Tilly processes, such as pg 155-163, with Python data objects (strings, list, etc.), building knowledge of what programmers do. It connects you closely to realistic outcomes of systems analysis and design work. And position you to learn quickly any systems anal. method. A final benefit of the Python OOP work is today's systems analysis, and design do a lot of work extracting and translating information. The result is challenging, but you will know more about it and how not to perform senseless internet searches looking for ideas.approaches to tackle it. **Tilley, Ch6: Overview** • The chapter does an excellent job detailing the components with little to no "geometric duds." • Notice by end of chapter everything you have done to this point is repeated here. Curious! **Python Training:** • By Wed you will be provided with customized training to support this work. It will have all that you need.

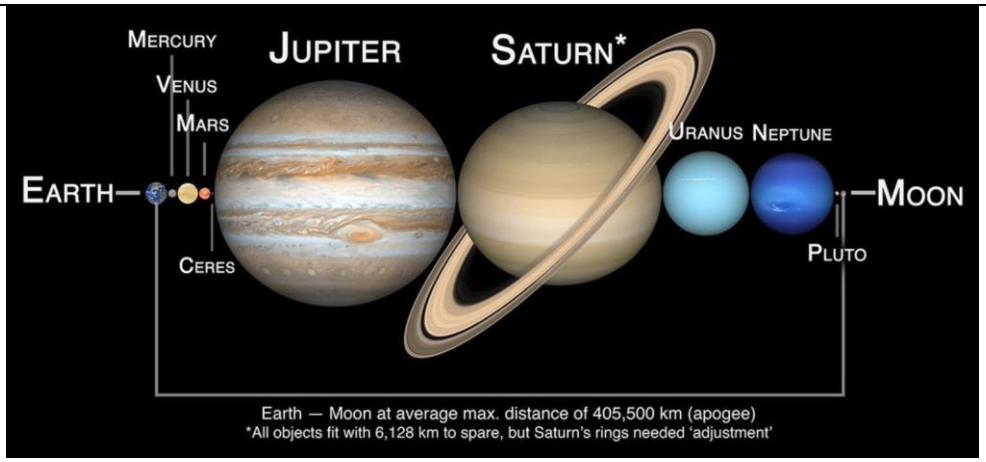
	 **ethics discussion text tilley p196** 	• Python crash course link below is good to reference and see examples for lists, loops, and similar. Feel free to dig into. • Real world python is super fun training exercises. **Other reference materials** • Matthes, E. (2019), Python Crash Course • Real world Python – FUN training examples • Matthes, Alien Invasion, Ch12. ○ Note: custom materials being provided replace Matthes chapters 1-11. Good to skim by priority: Ch:9,1,3,6 **Nothing due / Reading Only!** Class will start off discussing pg 196 ethics case study so please simply have your thoughts organized on that.

WK	Focus & Medium	Weekly Topic & Assignment
2.2 **9/9**	Focus / Goal lecture notes  **Michael Hammer acticle** 	**Goal:** wrap-up historical influence of business process reengineering • **lecture notes:** BPS's evolution with invention of machine learning and data warehousing. The institutionalized game changer of Amazon's kiva robotics **Ch5: data and process modeling** • data flow diagramming uses mostly an agreed upon set of symbols to represent processes, data flows, data stories and entities like transactions or physical items like a deposit ticket and goods. • the goal is to represent the information to be encoded by database programmers and develop apps that negotiate the transactions. • this class is less concerned on formality of box symbols but use circles to start and end a process, diamonds for decisions and rectangles for activities. • pg 153, agreed! try not to cross lines when building. • pg 155-159 does a nice job representing an actual system we could easily and realistic code for on hands-on python activities. • Unlike the book are goal is not to "write" about doing this work but actually code it using standard python data objects of lists, strings, dictionaries, tuples, and sets. a)**Reading:** Tilley, ch5, pgs 144-163 b)**Install** Python • Please watch video (i). The best course of action is installation via anaconda b.c it is engineered to auto-

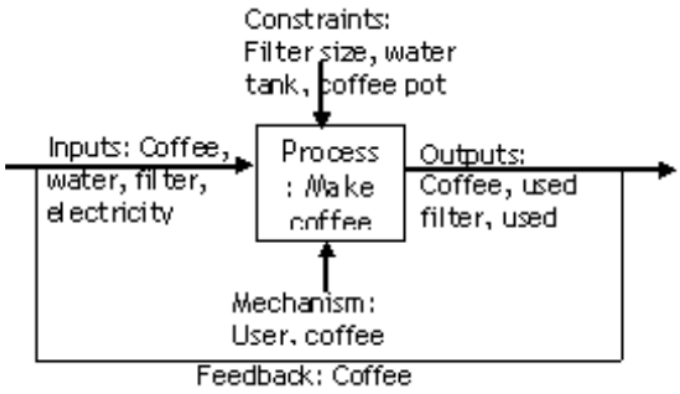
	Assignment
	
	A. Reading
	○ **Tilley,Ch5**
	B. Install Python
	
	Good luck w install!
		fix MANY challenges. However, if done wrong, the 1st time may take => 2-3x more work/time to fix. You "do not" have to figure this out yourself so please reach out with any questions.
		
		i. 1.3M views on YouTube: Install Anaconda Python, Jupyter Notebook And Spyder on Windows 10 - YouTube
		ii. good start place = jupyter notebook classic home
		iii. Jupyter :: Anaconda.org
		
		Python cloud
		• **online\cloud Jupyter Notebook**:
		  • online alternative – works great !
		  • https://jupyter.org/try-jupyter/lab/
		• JupyterLite — JupyterLite 0.1.0-beta.12 documentation

Wk	Focus & Medium	Weekly Topic & Assignment
2.1	Overview Podcast / Video Run videos at speed 1.25 Focus / Goal	**Ch2: Overview** ○ ch2 directs focus to business cases and how to identify a system for analysis. It augments learnings with factors contributing to project success/failure, purpose+ how.to a perform feasibility study, align priorities, and perform an preliminary investigation. ○ Section 2.9, "Preliminary Investigation" (p.26), outlines your revolving course focus building skills and techniques in ○ **Abstraction:** Which tool-kit model will help me quickly assess the situation asked of me? ▪ Quick assessments illustrate your ability to another party to grok salient factors, exercise skill by presenting a visual or data dashboard, and communicate back to manager or stakeholder. ▪ *Why should person X trust you*? Your responsible for building trust b/c it gets you access to more resources and what you need most, time. ○ **Data:** What data collection strategy will help me access inputs, outputs, resources, and constraints? ○ **Situational awareness:** After presenting initial response to business owner, what kind of model support, time, and resources do I have? Do I need? ✓ info.Tech resources usually can help get process metrics, source metric data, and any other information to meet your analysis goals. ✓ Data not what you need? Initiate estimation work.

	Model.2:SWOT Model.2:SWOT. Decision.Book perception... cartoon  	✓ Today, operations often have project planning documents associated with the system workflow you should inspect while applying your abstraction work. ✓ **SWOT.** When in doubt fall back to basics to help assess a situation's status with strengths, weaknesses, opportunities, and threats(tilley.45, krogerus.tschappelerp.12). **Perception & time <philosophy>:**  Figure 3. Illustrating how a hierarchy of specious presents and the passage of time may be represented by a sequence of compact dimensions in relative motion. (a) corresponds to SP_1, (b) to SP_2, (c) to SP_3, etc. link physical space, perceptual space, and memory ○ the course is not designed to dive deep into perception, time, and points of view. For systems modeling, learn to hone your logic representation skills ***and*** figure what you missed. ○ Do individuals experience time similarly? Does time affect perception? Quality of shared information?
Week	**Focus & Medium**	**Weekly Topic & Assignment**
1.2	Model.3: Swimlane IT Order Harmonization Example model.3.swimlane <bh.github> <how.to.doc> <wikipedia> sorry! in github you have to download to get link to work or use them here Artemis I	**Model.3.Swimlane** **Purpose:** use horizontal or vertical gradating color bars to demarcate business lines illustrating system inputs, activities, and decisions connected with arrows. **Assignment: Tilley Ch2 + Roughcut Swimlane diagram** ➢ Swimlanes no longer have notoriety as in 1993, and some IT professionals view them as a hindrance to what they need, that is, codified information. ➢ However, swimlanes are super at helping a senior manager or new employees quickly grasp what an organization is doing and how they are doing it. ➢ **"""You're the only resource, but you can have and do anything you want to do. Please include,"""** ➢ You're the only resource but can have, and do, anything you want to do. Please include, ✓ Square(ish) boxes to represent activities ✓ Lines to connect between activities ✓ Line arrowheads to show directionality between shapes ✓ Diamond(ish) boxes to represent decisions ✓ Text in squares + diamonds + on lines to detail happenings ✓ Optional: add a numeric index for each box & feel free to annotate "anyway" you like.

	[Space Launch System unmanned Moon mission](#)  **Swimlane** **Assignment request** **by 9/6 @6ish PM**	 **Example:** **Earth:Launch ↓** **Mars: Fuel up -> Open solar flares 3 yrs ↓** **Neptune: Turn into nano-space particulates** ❖ **Please email a .jpg, pdf however you build it.** ○ **File\SaveAs\often allows you select type .pdf** **-->'The goal is to be more thoughtful of your logic'<--**

Week	Focus & Medium	Weekly Topic & Assignment
1.1	Reading Podcast / Video What is business process re-engineering? **Run videos at speed 1.25** What is a system? inputs outputs resources constraints IDEF0 Handout	**Tilley, Ch 1.** [Intro to Systems Analysis](#) (free link) • 1st chapter is FREE !, use above link • Awareness & Design – Michael Hammer ○ https://www.youtube.com/watch?v=9oxM5JV7H50 • Business Process Re-engineering explained - ○ https://www.youtube.com/watch?v=v-jAf7L2Uak ▪ (10.5min/1.25=8.4min) • IBM Business process Analysis (6.5min/1.25=5.2min) ○ https://www.youtube.com/watch?v=1E6II2U1shY Utilize your abstraction instinct while reading because the name "EMS" isn't important, but the concepts are. https://www.niu.edu/ems/introduction/definition.html 1) definition is page 1 + 8 more pages using <next topic> 2) The EMS model 3) Benefits of EMS 4) Examples of EMS 5) Systems approach 6) Concept diagram <focus and perform abstraction here> 7) Processes, inputs, outputs a. Example of: inputs, outputs, resources, constraints 8) Summary • IDEFØ – Function Modeling Method – IDEF - website ○ 2nd example of input, output, res., constraint

	Assignment Request for 9/1 Assignment Example page Assignment example **Model.1:IDEF0**	Select a process you love or dislike. Define its input, outputs, resources, and constraints (IORC). Logically what goes into the system is either consumed or comes out. Notate ALL you think of. Then, list 5 to 10 high-level activities performed by the IORC. Use paper and pencil and send me a picture **anytime** end of the day tomorrow. I am only asking for a max of 15 min to whip up. Please spend more if having fun. Thank you for considering this fast turnaround, as I will use all work submitted to start Friday's lecture. Perform work as a team as desired or convenient. https://www.niu.edu/ems/introduction/constraints.html 

References

1. Kanigel, R. The One Best Way. Viking.

orders@ArtScroll.com

Morning to all Amazing Artscroll staff!
How are you today? It is raining heavy in Boston and I overslept thus late preparing my university lecture for my students today. So why am I writing this email?

I was blessed to have a Stone edition Artscroll English Tanach find me a little over a month ago. It has been an incredibly addition to my weekly studies and giving thanks for God in my life. My goal is to expand to daily prayers and for over a week I have been attempting to find the correct siddur. If someone has a few moments to think this through, perhaps they would be generous with their time and provide a couple recommendations. Or certainly only if it is that obvious and I am making things too complicated as usual!

At present I do not speak or read Hebrew but am going to learn so I think an interlinear version is the right first choice. Once I become familar I would then also get the transliterate to work on my pronounciation. I view my studying as a priority over pronounication so think I want an Hebrew/English siddur with the most explanations. Your offerings are amazing in this regard but now you have different contributors, newer versions, and so forth when expanded or focused commentary. How is one like myself to choose!? I was raised by my Jewish grandfather so I study very deeply so meaning, explanation, and background all help me form mental images and deepen my faith.

The last consideration is size. My current Bible study bag has room for one more hardback book in the smaller pocket size. I think pocket size is bigger than travel size and that is what I would want for

now. Since some books versions may be out of stock I would be willing to pay for more expensive hardback, such as the alligator, if appropriate. Again so many options!

thank you for considering all I have explained. I am sorry for this long email but I am VERY confident Artscroll can help me on this start. I know that I will be purchasing many more fresh books to make my own as my studies expand.

Thank you very kindly! ~brian.hogan