CS MENTORS PRESENT

ML WORKSHOP

DEC 4
4PM
CF 165

No machine learning background required, basic python recommended.

This will cover the basics of machine and deep learning, data processing and model creation / deployment.

FOR DISABILITY RESOURCES, CONTACT 360-650-3083.
Chains of Trust Workshop

with Austin Tipton

Austin attended WWU for two years and graduated with a degree in Computer Science. During this time he became interested in information security and specifically how bad actors use systems to gain unauthorized access. He currently works as a Security Engineer for Anvil Ventures.

Learn how chains of trust work with TLS and how to break that chain of trust with Burp Suite.

This will be a hands-on workshop. Please bring a Linux computer or Linux VM with OpenVPN and Java installed.

Before the workshop, please attempt the first challenge at hackthebox.eu/invite

360-650-2863

presented by:
WWU Cybersecurity Club

CSCWWU@gmail.com

wwucyber.com/about/
You’re invited to the Fall 2019
CS Study Break!

3 - 4 pm
Thursday, Dec 5th
CF 4th Floor Foyer

Take a break from all your hard work to enjoy some cookies, hot cocoa, and make paper snowflakes with friends!

For disability resources, contact 360-650-3083
Announcements

- Code/answers for Monday's quiz are linked on the course website.
- No A5's accepted after tomorrow night.
- Office hours 10-11:30 today as usual.
How to Study

Reading is not enough: **solve problems.**

- **Goals** slides: can you do these things? Try and see.

- **Terminology**: be able to discuss the meaning of all words that appear in blue in the slides.

- **Socrative questions**: make sure you know how to solve them. Then, try code in Thonny or compare answers with your peers.

- **Demo code**: solve the same problem without looking at my code.

- **QOTDs**: still available on Canvas - make sure you know how to solve them. *(please don't re-submit QOTDs)*

- **Sample coding problems** on Canvas

- **Exercises** from the eBook
What to study

• The final exam is cumulative.

• A comprehensive study guide can be generated by concatenating all Goals slides.
Hardware and interactions
Pseudocode / algorithms
Comments
Data types and conversions: int, float, str, bool
Function calls; arguments and return values
Variables
Math, comparison, and logical operators, precedence
Statement vs expression
Binary conversion
If/elif/else, nesting
While loop syntax and behavior
Importing modules
For loops, range
Defining functions with and without return values and parameters
Docstrings, specs, pre/postconditions
Local variables, variable scope; parameters are local variables
Tuples - unpacking, packing, return values and parameters
Function composition
Strings: operators, len, indexing, negative indices, slicing, in, lexicographic ordering
String methods: upper, lower, find, replace
Lists: same stuff as strings
Lists: modifying using assignment, append, extend, concatenation, insert, remove, del
Lists are mutable; variables hold references:
  Multiple variables can refer to the same object
  You can pass a reference to a mutable object into a function
Dictionaries: creation, assignment/indexing, in, del; iterating over keys and values
Files: open function, "r" vs "w"
  Read(), read(size), readlines(), write(string), seek(pos)
  Iterating over a file object

These are my notes on what to write problems about, generated from the goals slides.
Today

• Bonus Python ninja feature: list comprehensions
• A4 solution code
• Your questions
• Midterm Exam greatest hits
Friday (tentatively)

• Bonus Python ninja feature: f-strings
• Bonus Python WTF: recursive functions
• A5 solution code
• Your questions
List comprehensions
My A4 Solution
Midterm Exam: Greatest Hits

1pm edition

• 1.2: print's sep kwarg
• 1.3: input's return value
• 2.7: "5" * 3 + "22"
• 4: for m in ["A", "B"]
• 5: while num1 >= num2
• 12: avg and high score
Midterm Exam: Greatest Hits

9am edition

• 1.1 - print's return value
• 1.3 - input's return type
• 2.7 - "5" * 3 + "44"
• 4 - for m in ["T", "F"]
• 5 - while num1 >= num2
• 12 - avg/high score
• 2.1 - 6/2
• 3 - for x in range(5,-1,-3)