COURSE SCHEDULE

Week	Python and development Tooling	Software Development Practices & Processes	Due by START of class [R]: reading →: assignment to submit
1 **1/22**	INTRODUCTIONS Syllabus Installation: *Python 3, Editor, Git* Explore terminal & venv	INTRODUCTION TO AGILE Manifesto Subway Map	→Pre-test and skills survey →Install Python 3 [R] Syllabus [R] Beck et. al 2001 [R] Nerur & Balijepally 2007 [R] Lubanovic Ch. 1, 19 *(thru IDE section, pp. 409-416)* [R] Okken A1
2 **1/29**	TOOLING/ECOSYSTEM Version control: Git & Github Decentralized Flask *Assignments: Git & Github, Pair Exercise #1 (numbers, vars)*	PROJECT PLANNING Requirements Time estimation (planning poker) User stories & Tasks	→Flask App: Part 1 (basic app) [R] Lubanovic Ch 2-3, 18 (*thru flask section, pp. 375-397*) [R] P & M Ch 1-4 [R] Alwis & Silito 2009
3 **2/5**	PYTHON Data structures & Control flow *Pair Exercise #2 (strings, loop, lists, dict)*	DEVELOPMENT METHODS Method spotlight: Scrum Method spotlight: XP	→Git & Github exercise →Pair Exercise #1 (nums, vars) →Project Part A [R] Lubanovic Ch 4-8 [R] Kadenic et al 2023
4 **2/12**	WRITING CODE Functions, objects, and classes Import *Pair Exercise #3 (functions & classes)*	WHAT IS A PROGRAM?	→Pair Exercise #2 (loop, list, dict) →Flask App: Part 2 (login) [R] Lubanovic Ch 9-11 [R] Naur 1985
5 **2/19**	TESTING CODE Pytest	CODE DESIGN Principles: Dry, SRP, etc. Refactoring Unplanned tasks	→Pair Exercise #3 (funcs & classes) →Project Part B [R] Okken 1-3 [R] P & M Ch 5-6
6	DATA STORAGE	DESIGN PATTERNS	→Flask App: Part 3 (blog)

2/26	Data wrangling Files & directories Persistent storage	Midterm review	[R] Lubanovic Ch 12, 14, 16 [R] Okken Ch 4-5
7 **3/5**	MIDTERM	MIDTERM	NONE
8 **3/12**	Group Presentation #1		→Project Part C
SPRING BREAK WEEK 3/17 – 3/22			NONE
9 **3/26**	TEST CONFIGURATION Anatomy of a test in Python Pytest: fixtures & markers *In Class Team Exercise #1 (Code, Test)*	TEST-DRIVEN DEVELOPMENT Types of tests	[R] P & M Ch 7-8 [R] Okken 6-7, A3 [R] Janzen, D., & Saiedian 2005
10 **4/2**	CODE PERFORMANCE Timing Custom Decorators Concurrency Web scraping *Pair Exercise #4: concurrency* *Team "Client" Requests*	ITERATIONS AND BUGS Iteration design Bug handling	[R] P & M Ch 9-11 [R] Lubanovic Ch 15 [R] Larman & Basali 2003
11 **4/9**	CONTINUOUS INTEGRATION CI with Github actions: Build, Dependencies, Test Packaging & Deployment resources	FORMALIZING SOFTWARE DEVELOPMENT Continuous Integration (DEV) Continuous Delivery (OPS) DevOps	→Pair Exercise #4 (testing) [R] Okken A4 [R] P & M Ch 12 [R] Balalaie & Jamshidi 2016 [R] Chen 2015 [R] Lwakatare et al 2015 [R] Wettinger et al 2015
12 **4/16**	*In Class Team Exercise #2 (Refactoring)* Group work for presentation #2	AGILE REVISITED 2001 – 2023 Use & Effectiveness Team Experiences w/agile	[R] Kakar & Kakar 2023 [R] Lubanovic Ch 17 (Networking) [R] Thomas 2014, "Agile is Dead"
13 **4/23**	Group Presentation #2		→Project Part D

14 **4/30**	FUTURE OF CODING Tooling	FUTURE OF CODING Philosophy of coding	→Group Peer Evaluation [R] Somers 2023
15 **5/7**	WRAP UP	Future of coding exercise debrief Programming as Theory (revisit) Subway Map & Lessons Learned (in light of K&S 1995 – where would AI go on the sources of help chart?) Course Survey (Instructor: specifics of what worked) Course Evaluation (Dept for CGU purposes)	→Future of coding exercise [R] Kraut & Streeter 1995
16 **5/14**	FINAL EXAM		FINAL EXAM DUE