Department	International College of Liberal Arts		
Semester	Spring 2025	Year Offered (Odd/Even/Every Year)	Every Year
Course Number	DATA150		
Course Title	Introduction to Python Programming		
Prerequisites	None		
Course Instructor	PARIDA Abhishek	Year Available (Grade Level)	1
Subject Area	Data Science	Number of Credits	3
Class Style	Lecture	Language of instruction	English

(NOTE 1) Depending on the class size and the capacity of the facility, we may not be able to accommodate all students who wish to register for the course

Course Description	This course introduces Python programming for students with little to no prior programming experience. The course covers essential concepts, syntax, and common usage of Python. The focus is on hands-on exercises to reinforce the learned concepts. The course is designed for liberal arts students who want to learn programming as a tool for creative expression, data analysis, or digital humanities. Students will be provided with assignments and in-class exercises to enhance their understanding of the concepts taught in class.
Class plan based on course evaluation from previous academic year	Below are highlights in the course evaluation and some measures I intend to undertake in my future course delivery. Quite a few students pointed out the course was too fast for the beginners. The pace of the course can be controlled by repeating the materials. I constantly repeat when we have connecting topics; henceforth, I plan to keep aside a section of every lecture (say the last 10 minutes) to revise the material studied that day. Some students wrote the course is not a beginner-level course. I plan to take two lectures in the beginning to talk about the basics of computers and programming without writing a single piece of code; currently, I do this in one lecture. I also plan to do more manual exercises (manually executing the code on the whiteboard) and involve students' participation. This will make the class more engaging and help me monitor the students' understanding individually. Until now, I asked the students to share their code on Zoom, but from now on, I intend to devote time to the manual execution of the code. Apart from the above, I plan to repeatedly ask the students to write their questions to me on the Zoom chat and prepare the initial quizzes simpler.
Course related to the instructor's practical experience (Summary of experience)	N/A
Learning Goals	This course aims to introduce students to the fundamentals of programming using the Python programming language. Upon completing this course, students will have a solid foundation in Python programming and be able to write basic programs to solve problems in various domains.

iCLA Diploma Policy	DP1／DP2／DP4

iCLA Diploma Policy

(DP1) To Value Knowledge – Having high oral and written communication skills to be able to both comprehend and transfer knowledge

(DP2) To Be Able to Adapt to a Changing World – Having critical, creative, problem-solving, intercultural skills, global and independent mindset to adopt to a changing world

(DP3) To Believe in Collaboration – Having a disposition to work effectively and inclusively in teams

(DP4) To Act from a Sense of Personal and Social Responsibility – Having good ethical and moral values to make positive impacts in the world

Active Learning Methods	Problem-Based Learning				
More details/supplemental information on Active Learning Methods	N/A				
Use of ICT	Students must bring their laptops and install the Anaconda distribution of Python, which will be utilized for both classwork and homework assignments.				
Contents of class preparation and review	Students are advised to take handwritten notes; this will drastically increase their ability to retain the information. Further, they are expected to practice regularly. One to two hours of study is required before the class preparation, and an equal amount of practice is needed after each lecture.	Hours expected to be spent preparing for class (hours per week)	3 hours	Hours expected to be spent on class review (hours per week)	3 hours
Feedback Methods	The best way to correspond during the course is the UNIPA system or direct emails. Please check the UNIPA account regularly for updates related to classes. To have a better grade, be regular in the study, active and attentive in the class, do a revision of classwork regularly, and participate in-class quizzes.				

Grading Criteria		
Grading Methods	Grading Weights	Grading Content
Understanding of Concepts	30%	In-class participation, Homework Assignments, Class Quizes, Final Exam
Code Functionality	30%	Homework Assignments, Class Quizes, Final Exam
Attending Feedback Meetings	20%	Regular meeting after the assignment submission
Timeliness	20%	Homework Assignments

Required Textbook(s)	Eric Matthes- Python Crash Course: A Hands-On, Project-Based Introduction to Programming Al Sweigart- Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners
Other Reading Materials/URL	https://www.w3schools.com/python/ https://snakify.org/en/

Plagiarism Policy	Plagiarism is the dishonest presentation of others' work as if it were one's own. Duplicate submission is also treated as plagiarism. Depending on the nature of plagiarism, one may fail the assignment or the course. The repeated act of plagiarism will be reported to the University, which may apply additional penalties.
Other Additional Notes (Outline crucial policies and info not mentioned above)	N/A

(NOTE 2) Class schedule is subject to change

Class Schedule	
Class Number	Content
Class 1	Lecture 1 - Opening remarks and introduction to Python; Features of Python; Python philosophy; Why study Python?; Python Installation
Class 2	Lecture 2 - Python Basics: Arithmetic in Python; Variables and Strings, Expressions and Statements
Class 3	Lecture 3 - Variables; User input; f-string, Strings
Class 4	Lecture 4 - Booleans; Operators
Class 5	Lecture 5 - Python containers: Lists; Python loops: for; while
Class 6	Lecture 6 - Practice exercise on previously covered topics (string, list, booleans, operators, loops and others)
Class 7	Lecture 7 - Practice exercise on previously covered topics (string, list, booleans, operators, loops and others)
Class 8	Lecture 8 - Practice exercise on previously covered topics (string, list, booleans, operators, loops and others)
Class 9	Lecture 9 - Quiz 1
Class 10	Lecture 10 - Practice exercise on previously covered topics (quiz solution review, miscellaneous)

Class 11	Lecture 11 - Python containers: Lists; Tuples; Sets
Class 12	Lecture 12 - Comprehensions; Functions
Class 13	Lecture 13 - Practice exercise on previously covered topics (comprehensions, functions)
Class 14	Lecture 14 - Practice exercise on previously covered topics (miscellaneous)
Class 15	Lecture 15 - Quiz 2
Class 16	Lecture 16 - Practice exercise on previously covered topics (quiz solution review, miscellaneous)
Class 17	Lecture 17 - Python containers: Dictionaries; dictionary comprehensions
Class 18	Lecture 18 - Practice exercise on previously covered topics (miscellaneous)
Class 19	Lecture 19 - Practice exercise on previously covered topics (working with a dataset)
Class 20	Lecture 20 - Practice exercise on previously covered topics (working with a dataset)
Class 21	Lecture 21 - Quiz 3
Class 22	Lecture 22 - Practice exercise on previously covered topics (quiz solution review, miscellaneous)
Class 23	Lecture 23 - Lambda function; Map; Filter; Reduce; Zip
Class 24	Lecture 24 - Practice exercise on previously covered topics (lambda, miscellaneous)

Class 25	Lecture 25 - Exceptions and File I/O
Class 26	Lecture 26 - Practice exercise on previously covered topics (miscellaneous)
Class 27	Lecture 27 - Numpy and Matplotlib
Class 28	Lecture 28 - Practice exercise on previously covered topics (numpy and matplotlib)
Class 29	Lecture 29 - Quiz 4
Class 30	Lecture 30 - Practice exercise on previously covered topics (quiz solution review, miscellaneous)