SUMMER STRETCH 2024 SYLLABUS

Course Title:
Introduction to Python
Instructor(s):
Xuetao Ma
Bella Chang
Course Description:
This course offers a structured introduction to Python, one of the most in-demand programming languages. Students will start with the fundamentals of Python setup and basic scripting, advancing through key concepts such as data types, control structures, functions, and modules. The curriculum progresses to cover more advanced topics including data structures, file operations, error handling, and an introduction to object-oriented programming. Each session combines theoretical learning with practical, hands-on exercises, enabling students to apply their knowledge to real-world scenarios. The course culminates in a final project, where students will demonstrate their coding skills and creativity. Ideal for young learners with a keen interest in technology, this course aims to build a strong foundation in programming within an engaging and supportive environment.

Essential Questions:
1. What are the fundamental elements that make up a Python program?
2. How do Python's data structures optimize the management of data and enhance problem-solving capabilities?
3. In what ways can Python's control structures manipulate the flow of a program?
4. How do functions and modules promote code reusability and program organization?
5. What are the practical applications of file input/output operations in Python?
6. How can understanding exception handling improve the robustness and reliability of a program?
7. Why is object-oriented programming important, and how does it apply to Python?
8. What real-world problems can be solved by applying the concepts learned in this course?

Learning Outcomes:

At the end of the course, students will know/be able to:
(For example:

- ***Develop Python Programs:*** *Write and debug Python scripts using fundamental programming concepts like variables, loops, and conditionals.*
- ***Utilize Data Structures:*** *Effectively use lists, tuples, dictionaries, and sets to store, manipulate, and retrieve data.*
- ***Implement Functions and Modules:*** *Create reusable code and leverage Python's extensive library ecosystem for diverse applications.*
- ***Apply Object-Oriented Concepts:*** *Design and implement basic class-based applications.*
- ***Complete a Final Project:*** *Demonstrate comprehensive programming skills by planning, executing, and presenting a capstone project that solves a real-world problem.*

Student Assessment:

On the final transcript (Academic Record), the course grade will be given in letter form (A-B-C-D-F) and a percentage. However, a student can opt for a pass/fail grade for the course instead*: 70% or above is considered passing*. Final grades will not be rounded up. A student must let the instructor know before the last day of class (Wednesday, July 24) if they want the grade listed as pass/fail rather than a letter grade.

Breakdown:
30% Classwork and Participation
40% Homework
30% Presentations and Projects

Pass:

A+	A	A-	B+	B	B-	C+	C	C-
97-100	93-96	90-92	87-89	83-86	80-82	77-79	73-76	70-72

Fail:

D+	D	D-	F
67-69	63-66	60-62	0-59

Resources and Materials:
A laptop that can install and run anaconda is required.

Tentative Daily Schedule

(Below is an example of a daily schedule.)

Students are expected to be in class from 9AM to 2:20PM. We will follow a consistent (but not exactly the same every day!) daily schedule, which may look like the example below:

- 9AM - Morning check-in
- 9:15AM - Lectures and activities (15 min break around 10AM)
- 11:15AM - Morning recap, questions, announcements
- 11:30AM - Lunch break
- 12:30PM - Afternoon check-in
- 12:45PM - Study hall, group projects, office hours
- 2:20PM - Dismissal

Students may expect to spend some time outside the class time (9AM to 2:20PM) for individual studying—e.g. reviewing contents and lectures, preparing for weekly quizzes, and working on the final project.

Tentative Course Schedule
(Do not change the dates.)

Date	Topic(s)	In-class Activities
Monday, July 1	Introduction to Python & Setting Up	Install Python, write and run the first "Hello World" program, basic IDE introduction.
Tuesday, July 2	Variables, Data Types, and Basic Operations	Practice exercises with different data types and operators, simple calculation programs.
Wednesday, July 3	Control Structures: Conditional Statements	Implementing simple decision-making programs using if-else structures, interactive quizzes.
Monday, July 8	Implementing simple decision-making programs using if-else structures, interactive quizzes.	Writing loops to solve problems, creating simple games like guess-the-number.
Tuesday, July 9	Functions and Modules	Building custom functions, exploring Python standard libraries, simple projects using functions.
Wednesday, July 10	Data Structures: Lists and Tuples	Manipulating lists and tuples, creating a list-based project like a shopping list program.
Thursday, July 11	Data Structures: Sets and Dictionaries	Exercises with sets and dictionaries, building a simple dictionary-based app.
Monday, July 15	String Manipulation and Text Processing	String operations, creating a text-based game or a simple text analysis tool.
Tuesday, July 16	File I/O	Reading from and writing to files, creating a file-based data logging application.
Wednesday, July 17	Error Handling and Exceptions	Implementing try-except blocks, improving previous projects with error handling.
Thursday, July 18	Introduction to Object-Oriented Programming	Basic concepts of classes and objects, creating a simple class-based application.
Monday, July 22	Introduction to Projects	Team forming, project distributing, operating.
Tuesday, July 23	Project Development Day	Students start developing their final projects, with guidance and troubleshooting from the instructor.
Wednesday, July 24	Project Writing	Students continue to work on the projects.
Thursday, July 25	Final Project Presentation	Students finalize their projects, practice, and then present them to the class, other courses, and families.