ADVANCED TOPICS

The main goal of this unit is for students to begin to grapple with the essential questions below through learning more about how data fits into computer programs with Python. By the end of the week, students will be able to use and explain why and how dictionaries, data frames and objects are used to store and structure data. Additionally they will study and construct algorithms to make use of the data they are storing. They will be able to decide which structure to use based on the constraints of their project.

Students will learn about these concepts through three projects: Data Visualization, Sorting Algorithms and Social Network. Each project requires students to understand data structures to create different types of programs. The Data Visualization project teaches students about dictionaries, data frames and csvs while they create interactive graphs. The Sorting Algorithms project teaches students how to manipulate lists through nested loops. The Social Network project brings all of these ideas together while also introducing Object Oriented Programming. In the Social Network Project students create a social network structure (users, connections etc) using objects and subsequently learn how to incorporate algorithms and/or data analytics into this structure.

Unit Essential Questions	Unit Enduring Understandings
Who are Computer Scientists?	GWC girls are computer scientists! Computer scientists come from all different backgrounds. Diversity is important in CS because a diverse set of voices and perspectives improve us as people, as well as the products we create. Women were some of the earliest software designers
What is Computer Science? **What tools do computer scientists use to solve problems?**	Computer science is the study of solving problems using computational thinking, or by taking the steps needed to solve a problem and turning it into something a computer can follow. Girls explain why they chose a particular programming language, based on the problem they are solving. Girls explain why object oriented programming is powerful. There are many ways to store information in programs. Each has its own advantages and drawbacks.
How and why do we use computer science to solve problems?	Computer science is more than building programs - “doing” computer science requires problem solving. Have plan on paper before trying to code. You hardly ever get it working the first time, but keep trying!

How and why do we work with people to solve problems? **How do I work best?** **How can I communicate effectively with my teachers & peers?**	Working with different kinds of people can be hard, but it pushes you to think outside of your comfort zone, create better projects, and become a better person.

STUDENT OUTCOMES

Classroom Culture Goals:

- Students begin working more independently to ramp-up for final projects.
- Students struggle with hard problems in a safe environment to simulate more real-world problem solving. (Do a lot of code-brainstorms)
- Students get a taste of real technology jobs, including Data Science, Data Visualization Engineer, Data Analyst, Back-End Developer, and Algorithms.

Skill Goals

Students will **be able to:**

- Use dictionaries, lists, data frames and objects to store data and information in a program.
- Make decisions on how to structure data in a program that takes into consideration ease of use and adaptability.
- Give at least two examples of Women in computer science who use these data structures in their work.

Knowledge Goals:

Students will **know:**

- How to work with others to solve complex problems.
- How to manage their own frustration when code does not work initially.
- Planning for data structure before planning for functionality creates more adaptable code.

KEY EVIDENCE OF SUCCESS

Performance Task:

- Social Network Project
  - Students can choose to add either data visualization or algorithmic functionality to their social network project, which is based on an object-oriented structure.

Other Evidence:

- Journal entries
  A. How could you adapt whisker bot to be helpful around your house?
- Other projects

LEARNING PLAN

Lesson Plan and Overview	Daily Goal	Outputs and Look-Fors
Monday: Data Structures **Overview:** Students will learn new data structures, use them and be able to evaluate what data structures to use in what contexts. They will also learn about data scientists and data visualizers and what they do.	**Students understand:** • What a dictionary is and for what it is used. • The difference between a data frame and a dictionary. **Students can:** • Use dictionaries as a tool to solve problems. • Use data frames to store and structure data. • Use visualization library to create visualizations of data.	**Recipe Box** • Use Python dictionaries to store data. • Use terms "keys" and "values" correctly **Data Visualization Project** • Use Python and Pandas to look through and organize data • Use Python and Matplotlib to create data visualizations
Tuesday: Algorithims **Overview:** Today, students will learn about the concept of an algorithm. They will anchor their discussion with a simple math problem, and then go on to discuss and code a searching algorithm. Next, students experiment with Bubble Sort. The day ends with students getting practice completing a larger project in a group in order to learn more about algorithms and to independently manage group work.	**Students can:** • Search a sorted list using binary search. • Explain how bubble sort works. • Explain how algorithms are more than just code. • Explain an assigned algorithm using heuristics (pictures, actions, or a flow chart). • Explain how an assigned algorithm can be applied to a number of different real-world scenarios.	**Binary Search Program:** • Effectively uses pseudo code as a starting point for programming an algorithm. **Algorithm Presentations:** • Works effectively as a group to design and deliver a coherent and correct presentation about a technical topic.

Wednesday: OOP

Overview:

Today, students will learn about object oriented design and use this framework to create the structure for a simple social network. Students will end the day with a program that has limited functionality, but that they will expand upon on day 2.

Students understand:

- The difference between an object and an instance.
- Prioritizing how information is stored rather than the functionality of a program makes it more adaptable.

Students can:

- Create a class and declare an instance.
- Create an init method within a class.
- Declare class attributes.
- Use classes to structure data for the social network project.

User Class:

- Set up a class to store user information for social network.

Thursday: Social Network

Overview:

Today, students will add functionality to their social network by expanding upon their code from the previous day. They will work in groups to code at least two extensions per person. We end the day with a preview of final projects in order to allow students to hit the ground running with ideas on Monday.

Students can:

- Create a social network back-end that...
    - Stores users and their connections.
    - Prints a list of a user’s connections.
    - Finds the shortest path between two users through their connections.
    - Prints the degree of that shortest path.

Social Network Program:

- Students have added functionality to their social network program.
- Students can explain the power of object oriented programming

SUGGESTED CALENDAR

Monday: Data Structures

- Introduction - 20 min
- Explore data types - 40 min
- Suggested - Break/ Bonding/speaker (40 min?)
- Program a recipe - 60 min
- What a Data scientist does - 20 min
- Suggested - Lunch - 60 min
- Visualizing weather - 40 min
- Code Brainstorm - 30 min
- Further weather - 50 min
- Presentations - 15 min
- Reflection - 15 min
- Wrap up - 30 min

Tuesday: Algorithms

- Introduction - 30 min
- What is an Algorithm - 20 min
- Suggested - Break
- Introduction to Searching - Binary Search - 110 min
- Suggested - Lunch
- Introduction to Sorting - 25 min
- Suggested - Break/ Bonding
- Algorithm Research Project - 110 min
- Suggested - Break
- Reflection - 15 min
- Wrap-Up - 15 min

Wednesday: OOP

- Introduction - 30 min
- Object Oriented Design - 30 min
- Introduction to Objects: Classes and Attributes - 35 min
- Suggested - Lunch
- More on Objects: Methods - 25 min
- Network Class - 60 min
- Building a Testing Interface - 30 min
- Presentations - 30 min
- Reflection - 15 min
- Wrap-Up - 15 min

Thursday: Social Network

- Introduction - 20 min
- Session 1 - 60 min
- Suggested - Break
- Session 2 - 60 min
- Suggested - Lunch
- Gallery Walk - 30 min
- Preview final projects - 60 min
- Suggested - Break
- Reflection - 15 min
- Wrap-up - 15 min