Lesson 3: Reading Text Files

In this lesson, students will be introduced to the concept of text files and learn how to read data from them using Python. They will understand the importance of external data files in programming and explore real-life examples of their usage. The lesson will cover key methods such as `open()`, `read()`, and `readlines()` to read text files, and students will have the opportunity to practice their skills through guided and independent exercises. By the end of the lesson, students will have a solid understanding of how to work with text files and how it can enhance the functionality and efficiency of their programs.

## **Objectives:**

- Students will understand the purpose of external data files.

- Students will learn how to read data from an external text file.

## **Materials:**

- Computers with Python installed

- Text files with sample data (provided by the teacher)

- Projector or whiteboard

## **Bell-Ringer Activity (5 minutes):**

- Display a text file on the projector or whiteboard.

- Ask students to brainstorm and discuss the possible uses of text files in programming.

- After a few minutes, facilitate a class discussion and write down their responses on the board.

## **Introduction (10 minutes):**

- Explain to students that in this lesson, they will be introduced to text files and learn how to read data from them.

- Discuss the importance of external data files in programming and how they can be used to store and retrieve information.

- Provide real-life examples of situations where text files are commonly used, such as storing user data, reading configuration settings, or processing large datasets.

- Emphasize the relevance and practicality of learning how to work with text files in Python.

## **Direct Instruction (20 minutes):**

- Introduce the key methods used to read text files in Python: `open()`, `read()`, and `readlines()`.

- Explain that the `open()` function is used to open a text file and returns a file object.

- Demonstrate how to use the `read()` method to read the entire contents of a text file as a single string.

- Show an example of using the `readlines()` method to read the contents of a text file line by line and store them in a list.

- Discuss the advantages and disadvantages of each method and when to use them based on the specific requirements of a program.

## **Guided Practice (25 minutes):**

- Divide the class into pairs or small groups.

- Provide each group with a text file containing sample data.

- Instruct the students to write a Python program that reads the data from the text file and performs a specific task, such as calculating the average of a set of numbers or counting the occurrences of a certain word.

- Circulate around the classroom, providing guidance and support as needed.

- After the students have completed their programs, ask each group to share their code and explain how they approached the problem.

## **Independent Practice (25 minutes):**

- Assign two text file challenges to the students.

- Challenge 1: Write a Python program that reads a text file containing a list of names and sorts them in alphabetical order. Print the sorted list.

- Challenge 2: Write a Python program that reads a text file containing a series of sentences. Count the number of words in each sentence and print the total count.

- Allow the students to work individually on the challenges.

- Monitor their progress and provide assistance when necessary.

## **Exit Ticket (5 minutes):**

- Distribute an exit ticket to each student.

- Ask them to briefly explain the purpose of external data files and how to read data from a text file in Python.

- Collect the exit tickets before the end of the class.

## **Closure (5 minutes):**

- Review the main concepts covered in the lesson, including the purpose of external data files and the methods used to read data from text files in Python.

- Emphasize the importance of understanding how to work with text files in programming and how it can enhance the functionality and efficiency of a program.

- Encourage students to explore and experiment with text file manipulation in their future programming projects.

## **Common Core Standards:**

- CCSS.ELA-LITERACY.RI.9-10.1: Cite strong and thorough textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text.

- CCSS.ELA-LITERACY.RI.9-10.2: Determine a central idea of a text and analyze its development over the course of the text, including how it emerges and is shaped and refined by specific details; provide an objective summary of the text.

- CCSS.ELA-LITERACY.RI.9-10.3: Analyze how the author unfolds an analysis or series of ideas or events, including the order in which the points are made, how they are introduced and developed, and the connections that are drawn between them.