Lesson 4: String Handling Techniques

In this lesson, students will learn about string-handling techniques in Python. They will explore the `len()` function to find the length of a string and learn how to access individual characters using indexing. Through guided and independent practice, students will apply these techniques to manipulate strings and solve programming challenges. The lesson will also include a bell-ringer activity, a quiz, worksheets, and activities to reinforce the concepts learned. By the end of the lesson, students will have a solid understanding of string operators and be able to use for loops with string operations.

## **Objectives:**

- Students will be able to describe the function of string operators.

- Students will be able to use string-handling techniques.

- Students will be able to use for loops with string operations.

## **Materials:**

- Computers with Python IDE installed

- Projector or whiteboard

- Handouts with programming exercises

- Python documentation (online or printed)

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

1. Display the following code on the projector or whiteboard:

```python

word = "Hello"

print(len(word))

```

2. Ask the students to predict the output of the code.

3. Allow a few students to share their predictions.

4. Run the code and discuss the output with the students.

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

1. Explain to the students that today's lesson will focus on string-handling techniques in Python.

2. Define string-handling as the ability to manipulate and work with text data.

3. Discuss the importance of string-handling in programming and real-life applications.

4. Explain that they will learn two new techniques: finding the length of a string and accessing individual characters in a string.

5. Provide examples of how these techniques can be useful in various scenarios.

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

1. Introduce the `len()` function and explain that it returns the length of a string.

2. Demonstrate the usage of `len()` with different examples.

3. Discuss the common mistake of counting spaces as characters and clarify that spaces are included in the length count.

4. Introduce the concept of indexing and explain that each character in a string has a unique index.

5. Show how to access individual characters in a string using indexing.

6. Emphasize that indexing starts at 0 and explain the concept of zero-based indexing.

7. Provide examples of accessing characters at different positions in a string.

## **Guided Practice (20 minutes):**

1. Divide the students into pairs.

2. Distribute handouts with programming exercises related to string-handling techniques.

3. Instruct the students to work together to complete the exercises, using the `len()` function and indexing to manipulate strings.

4. Circulate the classroom to provide assistance and guidance as needed.

5. After the students have completed the exercises, review the solutions as a class, discussing different approaches and addressing any questions or misconceptions.

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

1. Explain to the students that they will now apply their string-handling skills to a programming challenge.

2. Present the "Guess the Word" game challenge:

- The program should randomly select a word from a given list of words.

- The player should be prompted to guess the word by entering a letter.

- The program should check if the guessed letter is in the word and provide appropriate feedback.

- The player should continue guessing until they correctly guess the entire word or run out of attempts.

- The program should keep track of the number of attempts and display a message indicating whether the player won or lost.

3. Allow the students to work individually on the challenge, using their knowledge of string-handling techniques and any additional Python concepts they have learned.

4. Encourage students to think creatively and use their problem-solving skills.

5. Monitor the students' progress and provide assistance as needed.

6. Once the students have completed the challenge, invite a few students to share their solutions with the class.

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

1. Distribute exit tickets to each student.

2. Ask the students to write a brief summary of the string-handling techniques they learned in today's lesson.

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

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

1. Recap the main points of the lesson, emphasizing the importance of string-handling techniques in programming.

2. Remind the students to practice and explore more string methods on their own.

3. Encourage the students to refer to the Python documentation for additional information and examples.

4. Thank the students for their participation and effort in the lesson.

**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.