Lesson 2: Representing Algorithms

In this lesson, students will learn about algorithms, code, and computer programs. They will understand how algorithms are step-by-step procedures or sets of rules for solving specific problems. Students will also explore how code is a set of instructions written in a programming language that can be executed by a computer to implement algorithms. They will learn about the importance of clear and precise code to ensure the desired outcome. Additionally, students will be introduced to flowcharts as a visual representation of algorithms and will practice creating their own flowcharts. By the end of the lesson, students will have a solid understanding of the differences between algorithms, code, and computer programs, and will be able to apply their knowledge to create flowcharts and write code.

## **Objectives:**

- Students will be able to differentiate between algorithms, code, and computer programs.

- Students will understand how algorithms can be represented as written descriptions, flowcharts, and code.

- Students will be able to identify and use flowchart symbols to develop a flowchart.

## **Materials:**

- Whiteboard or projector

- Markers or chalk

- Handouts with flowchart symbols

- Computers with programming software (optional)

## **Bell-Ringer Activity:**

1. Display a simple algorithm on the board: "How to make a peanut butter and jelly sandwich."

2. Ask students to think about the steps involved in making the sandwich.

3. Give students a few minutes to write down the steps individually.

4. Afterward, ask a few students to share their steps with the class.

## **Introduction:**

1. Begin by asking students if they have ever followed a recipe or set of instructions to complete a task.

2. Explain that algorithms are similar to recipes or sets of instructions that help us solve problems or complete tasks.

3. Define algorithms as a step-by-step procedure or set of rules for solving a specific problem.

4. Give examples of algorithms in everyday life, such as following a recipe, solving a Rubik's Cube, or solving a math problem.

## **Direct Instruction:**

1. Explain that code is a set of instructions written in a programming language that can be executed by a computer.

2. Discuss how code is used to implement algorithms and solve problems.

3. Show examples of simple code snippets and explain how they relate to algorithms.

4. Discuss the importance of writing clear and precise code to ensure the desired outcome.

## **Guided Practice:**

1. Introduce the concept of flowcharts as a visual representation of an algorithm.

2. Show examples of flowcharts and explain how they can represent the steps of an algorithm.

3. Provide handouts with flowchart symbols and their meanings.

4. Guide students through the process of creating a simple flowchart for a given algorithm, such as "How to brush your teeth."

## **Independent Practice:**

1. Assign students to create their own flowcharts for different algorithms, such as "How to tie a shoelace" or "How to make a cup of tea."

2. Encourage students to use the flowchart symbols they learned during the guided practice.

3. If computers are available, students can also translate their flowcharts into code using programming software.

## **Exit Ticket:**

1. Distribute an exit ticket to each student.

2. Ask students to write a brief explanation of the difference between algorithms, code, and computer programs.

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

## **Closure:**

1. Review the main points of the lesson, emphasizing the difference between algorithms, code, and computer programs.

2. Summarize the importance of clear and precise instructions in algorithms and code.

3. Encourage students to continue practicing their algorithmic thinking skills and coding abilities.

## **Common Core Standards:**

- CCSS.ELA-LITERACY.RST.9-10.3: Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.

- CCSS.ELA-LITERACY.RST.9-10.4: Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 9-10 texts and topics.

Exit Ticket:

1. Distribute an exit ticket to each student.

2. Ask students to write a brief explanation of the difference between algorithms, code, and computer programs.

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