Lesson Plan: Code Explorer - Your First Programming Adventure

Materials Needed:

- A computer with internet access
- A web browser (like Chrome, Firefox, or Safari)
- A notebook and pen/pencil for brainstorming and notes
- Access to the following free websites:
  - Scratch: scratch.mit.edu
  - Replit (for Python): replit.com
  - JSFiddle (for JavaScript): jsfiddle.net

Lesson Overview

This hands-on lesson introduces the foundational concepts of computer science and programming through exploration and creativity. Instead of just reading definitions, you will become a "Code Explorer," experimenting with different programming languages to see how they work and then building your very own mini-program.

Learning Objectives

By the end of this lesson, you will be able to:

- Define Computer Science and Programming in your own words, based on your experience.
- Identify and compare a block-based language (Scratch) with a text-based language (Python). (CRD-1.A.2)
- Explain that different programming languages are used to create programs and have different features. (CRD-1.A.1, CRD-2.B.1)
- Create a simple program that uses a variable to store and reference data. (AAP-2.A.2)
- Explain the basic difference between a language that is translated before running and one that is translated as it runs. (AAP-2.A.3)
- Identify a computing innovation and describe its purpose. (CRD-1.A.1)

Part 1: The Spark - Computing Innovations (10 minutes)

Activity: Innovation Investigation

1. **Brainstorm:** Open your notebook. List three of your favorite apps, video games, or websites. These are all **Computing Innovations**—they are programs that have changed how we play, connect, or learn.
2. **Discuss:** Think about one of them.
   - What problem does it solve or what purpose does it serve? (e.g., "Instagram helps me share photos with friends.")
   - Who do you think made it? How do you think they built it?

Teacher's Note: This connects the abstract idea of "Computer Science" to something tangible and relevant in your life. Everything you listed was built by people using a set of instructions. Let's find out how!

Part 2: The Explorer - What are Programming Languages? (25 minutes)

Activity: Three Languages, One Mission

Your mission is to make the computer display the message, "Hello, [Your Name]!" You will do this using three different languages. This will show you that while the goal is the same, the instructions can look very different.

1. **Language #1: Scratch (Block-Based)**
    - Go to the Scratch editor.
    - From the left-side menu, drag these blocks into the coding area and snap them together:
        1. Drag a `when green flag clicked` block (from the "Events" circle).
        2. Drag a `say Hello!` block (from the "Looks" circle).
        3. Click inside the word "Hello!" and change it to `Hello, [Your Name]!`.
    - Click the green flag above the cat sprite. Did it work? Congratulations, you just wrote a program!
2. **Language #2: Python (Text-Based)**
    - Go to the Replit Python editor.
    - In the middle coding panel, type this single line of code:
      `print("Hello, [Your Name]!")`
    - Click the green "Run" button at the top. Look at the output on the right. You did it again!
3. **Language #3: JavaScript (Text-Based)**
    - Go to JSFiddle.
    - In the JavaScript panel (bottom left), type this line of code:
      `alert("Hello, [Your Name]!");`
    - Click the "Run" button at the top left. A pop-up box should appear. Success!

Reflection:

- Which language was the easiest to understand just by looking at it? Why? (This touches on block vs. text-based languages).
- How was typing code in Python different from dragging blocks in Scratch?

Teacher's Note: You've just discovered that **Programming** is giving a computer instructions in a language it understands. **Computer Science** is the whole field of study—it includes programming, but also problem-solving, designing solutions, and understanding how computers work. You also saw that some languages (like Python) are often *interpreted* (translated and run line-by-line, like a live translator), while others are often *compiled* (the whole program is translated into machine code first, like translating a whole book before anyone reads it). For today, just know there are different ways computers process our instructions! (AAP-2.A.3)

Part 3: The Creator - Build Your Own Program! (45 minutes)

Activity: Choose Your Challenge

Now it's time to be creative. Choose **one** of the two challenges below and build it using either Python or Scratch (your choice!). This time, you'll use a **variable**—a container for storing information that can change.

Challenge A: "Mad Libs" Story Generator (Recommended for Python)

1. The goal is to ask the user for a noun, a verb, and an adjective, and then use their answers to tell a funny story.
2. You will need variables to store the user's answers. In Python, you can ask for input and store it like this:
   ```
   noun = input("Give me a noun: ")
   verb = input("Give me a verb: ")
   adjective = input("Give me an adjective: ")
   ```
3. Then, use the `print()` function to combine your text and the variables into a story. For example:
   ```
   print("The " + adjective + " " + noun + " decided to " + verb + " by the lake.")
   ```
4. Run your code and test it out! Try to write 2-3 sentences for your story.

Challenge B: "Personal Greeter" Interactive Character (Recommended for Scratch)

1. The goal is to make the cat sprite ask for your name and then use it to say hello and ask a question.
2. Use the `ask and wait` block from the "Sensing" category. Type `What's your name?` in the block.
3. The user's answer is automatically stored in a special variable block called `answer`.
4. To use the answer, you'll need the `join` block from the "Operators" category. This lets you combine text.
5. Combine a `say` block with a `join` block to make the cat say something like: `join (Hello, ) (answer)`
6. Try to make your character ask a second question after greeting the user by name!

Teacher's Note: In both challenges, you used variables (`noun`, `answer`) to store information. Notice how the program refers to the variable's name to get the data inside it. This is a fundamental concept in all programming! (AAP-2.A.2)

Part 4: The Showcase - Reflection and Wrap-Up (10 minutes)

Activity: Show and Tell

1. Demonstrate your finished program. Run it and show what it does.
2. Answer these questions:
   - Which language did you choose for your project, and why?
   - What is a variable? How did you use one in your project?
   - In your own words, what is programming?
   - What was the most fun part of this lesson? What was the most challenging?

Extensions & Deeper Dives (Optional)

- **For the Mad Libs project:** Add more variables! Ask for a place, a food, a number, etc., and make your story even funnier and more complex.
- **For the Personal Greeter project:** Use an `if/else` block (from "Control") to make the character say different things based on the user's answer to your second question.
- **Research:** Pick one of the apps you listed in Part 1 and search for "what programming language is [app name] written in?". You'll be surprised at the variety!