```html

# Lesson Plan: Roblox Game Design 101 - From Code to Creation

### Materials Needed:

- A computer (Windows or Mac) with an internet connection.
- A free Roblox account.
- Roblox Studio installed (it's a free download with a Roblox account).
- A notebook or document for brainstorming and taking notes.

---

### 1. Learning Objectives

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

- Define computer science, programming, and programming languages in your own words.
- Identify Roblox as a "computing innovation" and brainstorm new ways to use its tools.
- Create a simple script in Roblox Studio using the Luau programming language.
- Use a variable to control a game object (a "Part") and understand why clear variable names are important.
- Apply your knowledge to build a working, interactive element: a disappearing platform!

### 2. Alignment with Standards (AP Computer Science Principles)

- **CRD-1.A.1:** A computing innovation is a physical, non-physical software, or conceptual solution that includes a computer or program as an integral part of its functionality.
- **CRD-1.A.2:** A computing innovation can be an existing technology that is used in a new way.
- **CRD-2.B.1:** A program is a set of instructions for a computer to execute.
- **AAP-2.A.2:** The way variables are used in a program can change based on their interaction with other parts of the program.
- **AAP-2.A.3:** The name of a variable should describe the data it contains or its purpose.

---

## Lesson Activities

### Part 1: The Spark - What is All This Stuff, Anyway? (10 minutes)

Let's start by connecting what you love to the big ideas of technology.

1. **Discussion:** What's your favorite game on Roblox? What makes it fun? Is it the gameplay, the story, the look? We're going to learn the secret behind how all of that is made.
2. **Big Ideas Introduction:**
    - **What is Computer Science?** Think of it as the master art of problem-solving using a computer. If the problem is "How do I make a character jump?", computer science is the whole process of figuring that out.
    - **What is Programming?** This is the act of writing the instructions. It's like writing a very specific recipe for the computer to follow. Our recipe today will be for a "Disappearing Platform." (**Standard CRD-2.B.1**)
    - **What are Programming Languages?** This is the language you write the recipe in. You can't just write "Hey computer, make this block disappear!" You have to use a special
```

language it understands. Roblox uses a language called **Luau**.

Part 2: The Idea - Roblox as an Innovation (10 minutes)

Let's think like a game developer.

1. **Define Computing Innovation:** A "computing innovation" is just a fancy term for a tool or idea that uses a computer program to do something cool. Roblox itself is a massive computing innovation! It's a platform for creating and sharing games. (**Standard CRD-1.A.1**)
2. **Brainstorming Challenge:** An innovation can also be using an existing tool in a new way. (**Standard CRD-1.A.2**) Let's brainstorm. Roblox gives us blocks, scripts, and characters. How could we use these tools in a brand new, creative way?
    - *Example:* Could we use blocks that change color based on music to create a dance floor?
    - *Your Turn:* Think of one cool, new idea for a Roblox game feature. What would it do? (e.g., a pet that follows you and copies your dance moves, a paintbrush that lets you paint new obstacles into the world).

Part 3: The Creation - Let's Build a Disappearing Platform! (30 minutes)

Time to get our hands dirty in Roblox Studio! This is where we apply everything we've talked about.

Step 1: Set Up Your World

1. Open Roblox Studio.
2. Click "New" and choose the "Baseplate" template. This gives you a nice open space to work.
3. In the top menu, go to the "Part" menu and click the block icon to insert a new Part. This will be our platform.
4. Use the "Scale" and "Move" tools to make it bigger and position it slightly above the ground so your character can jump on it.
5. In the "Explorer" window on the right, find your "Part" and rename it to something clear, like "DisappearingPlatform".

Step 2: Add a Script (The "Brain")

1. In the "Explorer" window, hover over your "DisappearingPlatform" and click the little "+" icon that appears.
2. From the menu, select "Script". This will open a new window where we can write our code.

Step 3: Write the Code (The "Magic Spell")

Delete the default "print('Hello world!')" text and carefully type the following code. We'll go over what each line means.

```
-- This variable will hold our platform part.
local platform = script.Parent

-- This is an endless loop so the platform keeps working.
while true do

    -- Step 1: Make the platform solid and visible
    platform.Transparency = 0
    platform.CanCollide = true
    wait(3) -- Wait for 3 seconds

    -- Step 2: Make the platform invisible and non-solid
```

```
    platform.Transparency = 1
    platform.CanCollide = false
    wait(3) -- Wait for 3 seconds

end
```

Code Breakdown:

- `local platform = script.Parent`: We're creating a **variable** named "platform". A variable is just a container for information. We gave it a clear name so we know exactly what it holds: the Part our script is attached to! (**Standard AAP-2.A.3**)
- `while true do ... end`: This creates an infinite loop, so our platform will disappear and reappear forever.
- `platform.Transparency = 0`: We are accessing our variable ("platform") and changing a property. Transparency of 0 means it's fully visible. (**Standard AAP-2.A.2**)
- `platform.CanCollide = true`: This makes the platform solid so you can stand on it.
- `wait(3)`: Pauses the script for 3 seconds.
- `platform.Transparency = 1`: A transparency of 1 makes it completely invisible!
- `platform.CanCollide = false`: This makes the platform non-solid, so you'll fall right through it.

Step 4: Test Your Game!

1. Click the "Play" button in the top menu.
2. Walk your character over to the platform. Watch it appear and disappear! Try to stand on it and see what happens when it vanishes.
3. Congratulations, you just wrote a working game program!

Part 4: The Debrief - What Did We Accomplish? (5 minutes)

- **Review:** How did we use a "program" (our script) to control a game object? How was our "variable" (`platform`) essential for making this work?
- **Reflection (Exit Ticket):** In your notebook, answer this: How is the disappearing platform script a small "computing innovation" within your game? How does it change the way a player interacts with the world?

Differentiation and Extension

Choose your own adventure for what to do next!

- **Support:** If you're having trouble with the code, double-check spelling and capitalization—they matter! If needed, you can copy and paste the code block to make sure it's perfect.
- **Challenge (Pick one!):**
  - **Time Master:** Can you change the `wait()` times? Make it disappear faster than it reappears.
  - **Color Wizard:** Can you make the platform change color right before it disappears as a warning? (Hint: Look for the `BrickColor` property. You might add a line like `platform.BrickColor = BrickColor.new("Really red")`).
  - **The Duplicator:** Can you copy the platform and its script to create a challenging obstacle course of multiple disappearing platforms?

Assessment

Your success on this lesson is based on creation and understanding, not a test!

- **[ ] Formative (Check-ins):** Answered questions during the discussion and brainstorming parts.
- **[ ] Summative (Final Product):** Successfully created a working disappearing platform in Roblox Studio that you can demonstrate.
- **[ ] Summative (Reflection):** Wrote a thoughtful answer to the "Exit Ticket" question, connecting your creation back to the idea of a computing innovation.

```
```