Lesson Plan

Lesson 1:	Learning Sequence
Lesson 2:	Learning Branch, Jump ( ), goto ( )
Lesson 3:	Making decisions, Conditional – if ( ) then else ( )
Lesson 4:	Fixing Errors, Bug and Debugging
Lesson 5:	Looping with repeat, bounded loops
Lesson 6:	Understanding Functions

Bonus lessons

Lesson A:	Introducing operations, greater, less than. Boolean - TRUE, FALSE
Lesson B:	Introducing Variable, string and numeric
Lesson C:	Nested repeat - Loop in Loop

Lesson 1: Learning Sequence

Before you start – If you haven't already, please read the CoderBunnyz Rulebook page 1-8 to be introduced to playing the game.

Lesson Overview

Students will do an introductory worksheet. Then they will play the sequencing level of CoderBunnyz. Finally, they will write their code as an algorithm.

Lesson Objective

- First students will do an introduction worksheet to introduce the concept of sequencing.
- Then they will play a level of CoderBunnyz, and arrange their code cards as a sequence of steps
- Finally they will count the number of code cards used as instructions and write/draw the sequence of cards taken. This is called algorithm writing.

Materials Needed

- Sequence worksheet(on the next page), board game, algorithm sheet 1.1, pencil

Getting Started

- Instructor will explain worksheet 1. Players will do the exercise.
- After the worksheet is complete, arrange the game (see Rule Book page 9, Level 1.1), explain the cards, movements, and rules. Choose the destination and get ready to start the game.

Activity

- Play level 1.1 of CoderBunnyz to program your bunny to eat their colored carrot and reach the destination. Continue till all players reach the destination.

- Then each player will review their code cards. That’s the sequence of code they will write on their algorithm sheet.
- Count the number of cards used to reach the destination and write those on the sheet 1.1. Also write the algorithm of the game played.

Fun Fact

The first computer programmer was a female, named Ada Lovelace.

1.Sequence

Series of ordered actions or steps

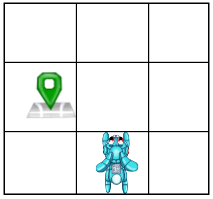
Read aloud

Sequence for playing a game when in a group

- Decide on the game
- Share the rules
- Start playing

Read aloud

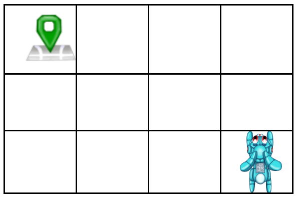
An example of sequence (Forward is in direction of bunny's eyes)



↑ ↰ ↑

Read aloud

Your turn to help bunny , use ↑ ↰ ↱



Practice Exercise ____ ____ ____ ____ ____ ____

In 5 steps or less create a sequence of your morning routine

1)
2)
3)
4)
5)

Practice Exercise

Write
Algorithm

1.1 Sequencing

Look at your code: Count how many code-cards you use in your coding today:

_____ MoveForward ( ); ⇧

_____ TurnLeft ( ); ↰

_____ TurnRight ( ); ↱

⇧↱ …

MoveForward ( );
TurnRight ( );
..

Using the symbol or text Write your Code for the game played!!

Sequencing in real programming languages?

The “sequence structure” is build into all the programming languages. Including Java, Python, C, C++ and others. The computer executes the statements one after the other in the order the code is written. This order is called the sequence.

Here’s a simple code example in Python updating 2 variables x and y in sequence.

```
1. x = 2
2. y = x + 1
3. y = 2*y
4. x = y - x
5. print(x, y)
```

Code Line	x	y	Comment
1	2	-	
2	2	3	3=2+1, using the value of x from the last line
3	2	6	6=2*3 on the right, use the value of y from the last line
4	4	6	4=6-2 on the right, use the value of x and y from the last line
5	4	6	print: 4 6

Here’s another code example in C++ printing statements in sequence-

```
int main ()
{
  std::cout << "Hello World! ";         // prints Hello World!
  std::cout << "I'm a C++ program;";    // prints I'm a C++ program
  std::cout << "Learning Sequencing;"; // prints Learning Sequencing
  std::cout << "Thanks CoderBunnyz!";  // prints Thanks CoderBunnyz
```

When program runs it prints the statements one by one in sequence!

```
Hello World!  I'm a C++ program; Learning Sequencing; Thanks
CoderBunnyz!
```