



# Overview:



# Table :

| All exercises          |        |  |             |
|------------------------|--------|--|-------------|
| Exercise               | Levels | Concepts   | Blocks Used |
| Fun with Basics        | 0/10   | Sequence, Algorithmic Thinking   | 0           |
| Loopy Loops            | 0/12   | Loops, Debugging   | 0           |
| Conditional Crops      | 0/12   | Conditional Statements, Pattern Recognition  | 0           |
| Backyard Functions     | 0/10   | Functions, Variables, Events   | 0           |
| Dog and the loops      | 0/8    | Loops, Variables, Functions  | 0           |
| Gardening Conditionals | 0/6    | Functions, Conditional Statements, Sequence, Algorithmic Thinking  | 0           |
| Swamp conditionals     | 0/4    | Conditional Statements, Loops, Variables, Sequence, Events, Functions, Decomposition, Algorithmic Thinking | 0           |
| Baloon pop functions   | 0/8    | Conditional Statements, Loops, Variables, Sequence, Events, Functions, Decomposition, Algorithmic Thinking | 0           |

|                               |     |  |   |
|-------------------------------|-----|--|---|
| Loops and castles             | 0/8 | Loops, Variables, Functions  | 0 |
| Desert conditionals           | 0/4 | Conditional Statements, Loops, Variables, Sequence, Events, Functions, Decomposition, Algorithmic Thinking | 0 |
| Predator bird functions       | 0/7 | Conditional Statements, Loops, Variables, Sequence, Events, Functions, Decomposition, Algorithmic Thinking | 0 |
| Functions on the field        | 0/9 | Conditional Statements, Loops, Variables, Sequence, Events, Functions, Decomposition, Algorithmic Thinking | 0 |
| Fun with Basics               | 0/3 | r  | 0 |
| Loopy Loops                   | 0/4 | r  | 0 |
| Conditional Crops             | 0/4 | r  | 0 |
| Backyard Functions            | 0/6 | r  | 0 |
| Fun with Basics - Grade 1 & 2 | 0/8 | .  | 0 |
| Loopy Loops - Grade 1/2       | 0/8 | .  | 0 |

## List of Concepts:

### Decomposition

Breaking down a problem into smaller, more manageable parts.

[Computational Thinking Concepts](#)

### Pattern Recognition

Identifying similarities or patterns within problems.

[Computational Thinking Concepts](#)

### Abstraction

Simplifying complex problems by focusing on essential details and ignoring unnecessary information.

[Computational Thinking Concepts](#)

### Algorithmic Thinking

Developing step-by-step instructions or rules to solve a problem.

[Computational Thinking Concepts](#)

## Sequence

Understanding and writing instructions in a specific order.

[Programming Concepts](#)

## Variables

Introducing the concept of containers for storing information.

[Programming Concepts](#)

## Loops

Repeating a set of instructions multiple times.

[Programming Concepts](#)

## Conditional Statements

Making decisions in the program based on certain conditions.

[Programming Concepts](#)

## Events

Reacting to user inputs or specific occurrences in the program.

[Programming Concepts](#)

## Functions

Creating reusable blocks of code to perform specific tasks.

[Programming Concepts](#)

## Data Types

Introducing the idea of different types of data, such as numbers, text, and Boolean values.

[Programming Concepts](#)

## Input and Output

Understanding how programs receive information (input) and produce results (output).

[Programming Concepts](#)

## Debugging

Identifying and fixing errors or mistakes in the code.

[Programming Concepts](#)

## Comments

Adding explanations and notes within the code for better understanding.

[Programming Concepts](#)

## Event Handling

Responding to events triggered by user actions or other parts of the program.

[Programming Concepts](#)

## **Graphics and Animation**

Introducing basic concepts of drawing and creating movement in a program.

[Programming Concepts](#)

## **Simulation**

Creating virtual scenarios to model real-world situations.

[Programming Concepts](#)

## **Collaboration**

Encouraging teamwork and sharing of code with others.

[Programming Concepts](#)

## **Iteration**

Repeating a set of instructions or a process.

[Programming Concepts](#)