# Introduction to Algorithmic Thinking and Informatics

\*\*Course Duration\*\*: 20 weeks  
\*\*Session Timing\*\*: Tuesdays, 8:30 AM - 9:10 AM  
\*\*Target Audience\*\*: 11-year-old pupils (Grade 5, French school system)  
\*\*Course Instructor\*\*: Mr. Mæhdi MOKHTARIIRANI

## Course Objectives

1. Introduce fundamental algorithmic concepts.  
2. Develop problem-solving skills and logical reasoning.  
3. Use Scratch as a tool to implement algorithms and visualize their behavior.  
4. Build familiarity with basic algorithms (sorting, encryption) and informatics principles.

## Weekly Breakdown

### Weeks 1–2: Introduction to Algorithms and Informatics

* - Understand what an algorithm is and how it is used in everyday problem-solving.
* - Discussion: 'What is an algorithm?' Examples from daily life (e.g., making a sandwich, tying shoes, changing a lightbulb).
* - Activity: Write step-by-step instructions for brushing teeth.
* - Exercise: Arrange shuffled steps for 'How to make hot chocolate' into the correct sequence. Solution provided.

### Weeks 3–4: Representing Algorithms with Flowcharts

* - Learn to represent algorithms visually with flowcharts.
* - Explain flowchart symbols: Start/End, Process, Decision, Arrow.
* - Class Activity: Create a flowchart for deciding 'What to wear based on the weather.'
* - Exercise: Create a flowchart for 'What to do if you’re hungry.' Solution provided.

### Weeks 5–6: Introduction to Scratch and Basic Programming

* - Learn Scratch basics and create simple programs.
* - Introduction to Scratch interface: Blocks, canvas, and sprites.
* - Class Activity: Make the Scratch cat say 'Hello World.'
* - Exercise: Program the Scratch cat to introduce itself and move 10 steps. Solution included.

### Weeks 7–8: Conditions ('If Statements')

* - Understand decision-making in algorithms.
* - Explain conditional logic with examples (e.g., 'If it rains, take an umbrella.').
* - Class Activity: Write a flowchart to decide 'What to do if it's a school day vs. the weekend.'
* - Exercise: Program a Scratch project where the cat says 'Good Morning' if it’s before noon and 'Good Evening' otherwise. Solution included.

### Weeks 9–10: Loops

* - Introduce repetition and loops in algorithms.
* - Explain loops: Real-life analogy (e.g., brushing teeth for 2 minutes = repeat steps).
* - Class Activity: Create a flowchart for 'Tying 5 balloons for a party.'
* - Exercise: Program the Scratch cat to say numbers 1 to 5 using a loop. Solution provided.

### Weeks 11–12: Combining Conditions and Loops

* - Learn how to use conditions inside loops.
* - Class Activity: Write a flowchart for 'Counting only even numbers from 1 to 10.'
* - Exercise: Program Scratch to make the cat say all even numbers from 1 to 10. Solution provided.

### Weeks 13–14: Introduction to Sorting Algorithms

* - Understand and implement basic sorting algorithms (focus on bubble sort).
* - Explain sorting: Using cards with numbers (e.g., arrange cards from smallest to largest).
* - Class Activity: Demonstrate bubble sort using physical cards.
* - Exercise: Program Scratch to sort three numbers entered by the user. Solution provided.

### Weeks 15–16: Introduction to Basic Encryption

* - Learn simple encryption techniques (e.g., Caesar cipher).
* - Explain encryption: Using examples (e.g., shift letters by 3).
* - Class Activity: Encrypt and decrypt a short message using the Caesar cipher on paper.
* - Exercise: Program Scratch to encrypt a word using a Caesar cipher. Solution provided.

### Weeks 17–18: Problem Solving and Algorithm Design

* - Develop problem-solving skills through real-life inspired scenarios.
* - Class Activity: Write an algorithm to divide a chocolate bar fairly among 4 friends.
* - Exercise: Program Scratch to calculate the remainder when dividing two numbers. Solution provided.

### Weeks 19–20: Review and Final Project

* - Apply knowledge to a creative project.
* - Class Discussion: Review key concepts (flowcharts, Scratch, conditions, loops, algorithms).
* - Final Project: Create a Scratch program for a game or interactive story that includes: at least one condition and one loop.
* - Exercise: Debug a provided Scratch program with errors. Solution provided.