Welcome to this CoGrammar Lecture: Introduction to Algorithms and Problem-Solving

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

CoGrammar

Software Engineering Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** throughout this session, should you wish to ask any follow-up questions.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: Questions

Software Engineering Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support
- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: Feedback on Lectures

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Enhancing Accessibility: Activate Browser Captions

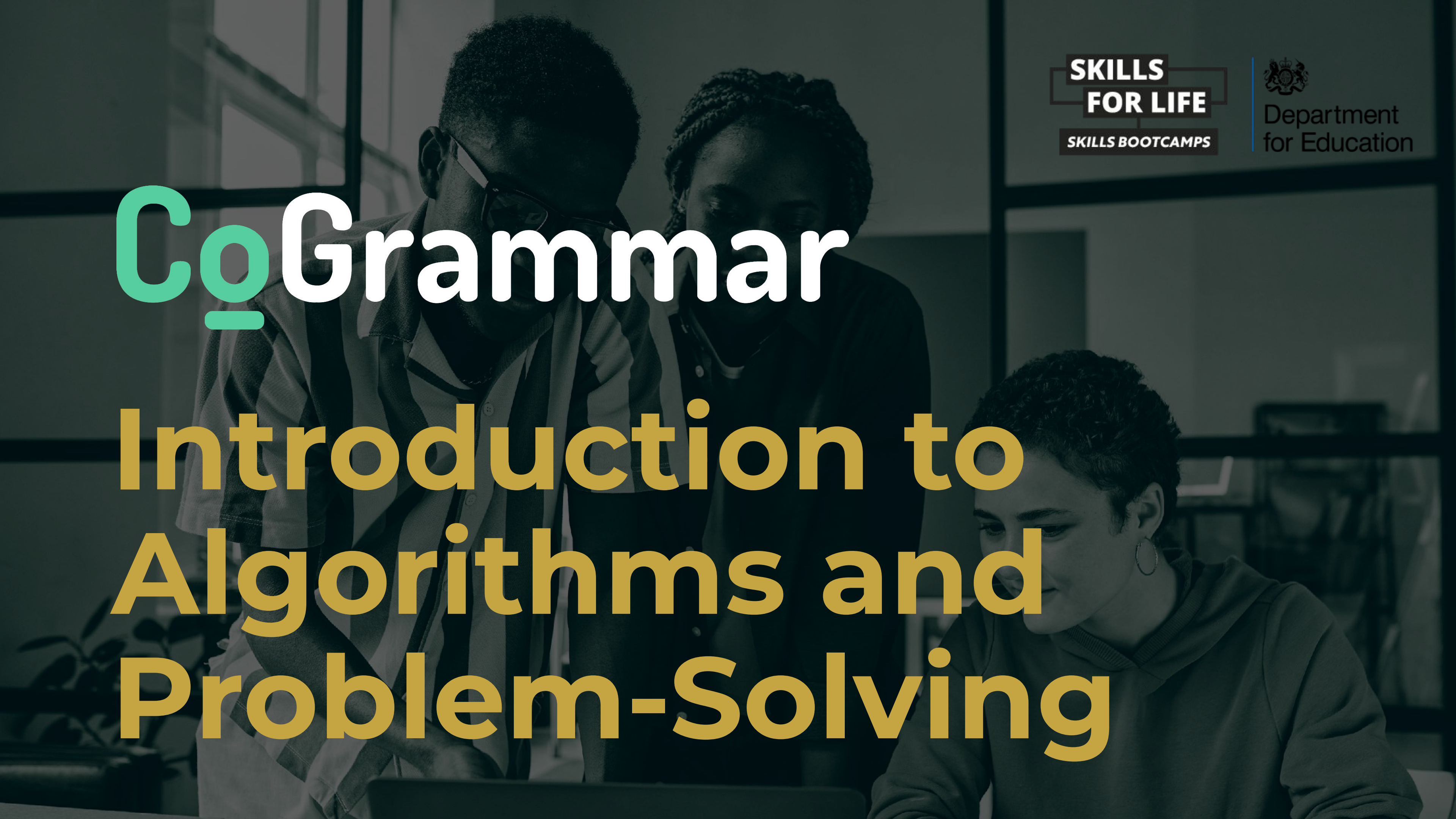
Why Enable Browser Captions?

- Captions provide **real-time text for spoken content**, ensuring inclusivity.
- Ideal for individuals in noisy or quiet environments or for those with **hearing impairments**.

How to Activate Captions:

1. **YouTube or Video Players:**
   - Look for the CC (Closed Captions) icon and click to enable.
2. **Browser Settings:**
   - Google Chrome: Go to *Settings > Accessibility > Live Captions* and toggle ON.
   - Edge: Enable captions in *Settings > Accessibility*.

CoGrammar



SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

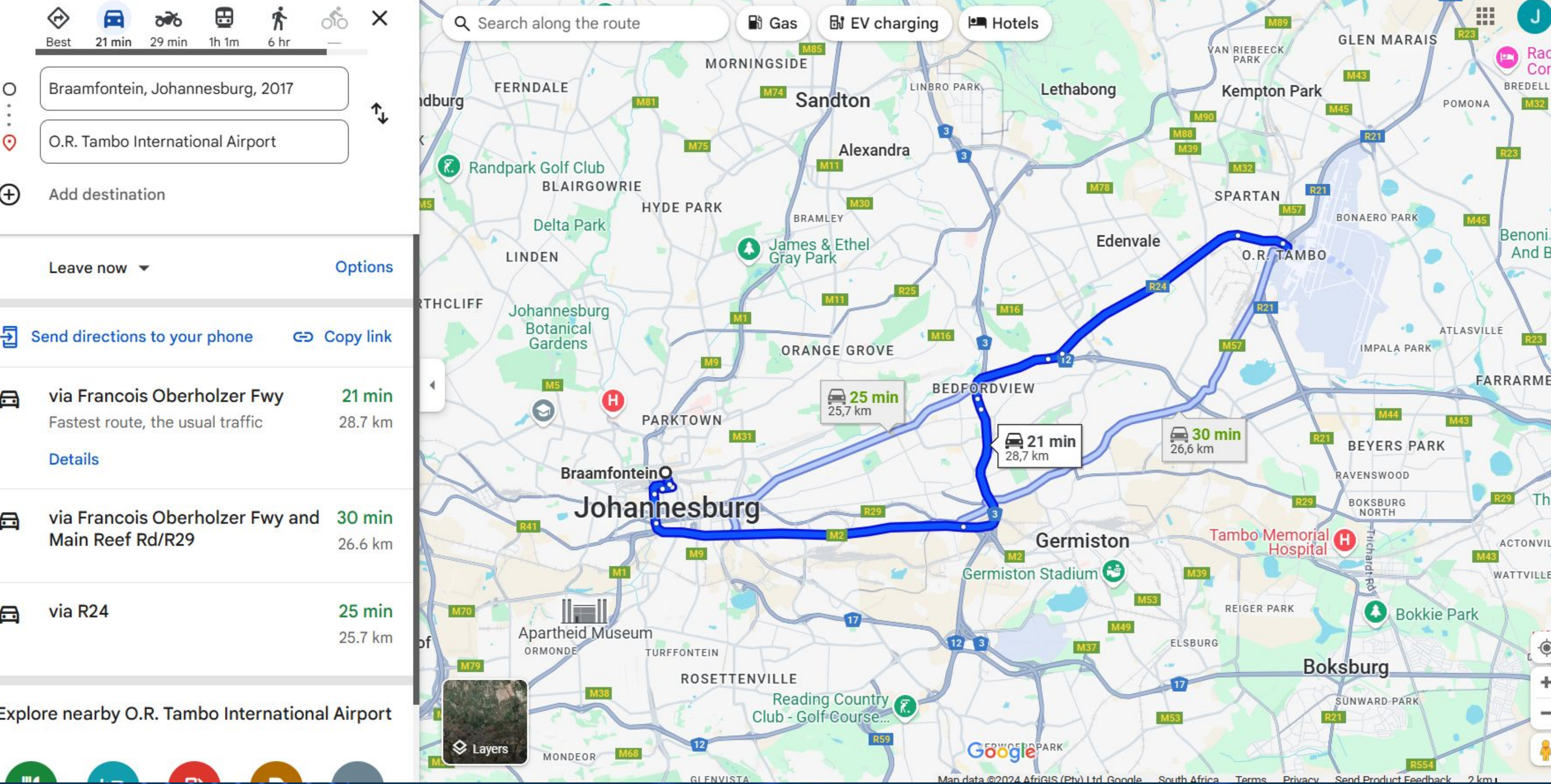
Introduction to Algorithms and Problem-Solving

Learning Objectives & Outcomes

- Define what an algorithm is and why it is important.
- Recognize the link between problem-solving and algorithms.
- Break problems into manageable parts.
- Identify and use the three types of instructions: sequence, conditional statements, and loop statements.
- Apply problem-solving skills to design a simple algorithm for navigating a maze.

Building a Foundation for Logical Thinking

CoGrammar



Braamfontein, Johannesburg, 2017

O.R. Tambo International Airport

Add destination

Leave now

Options

Send directions to your phone

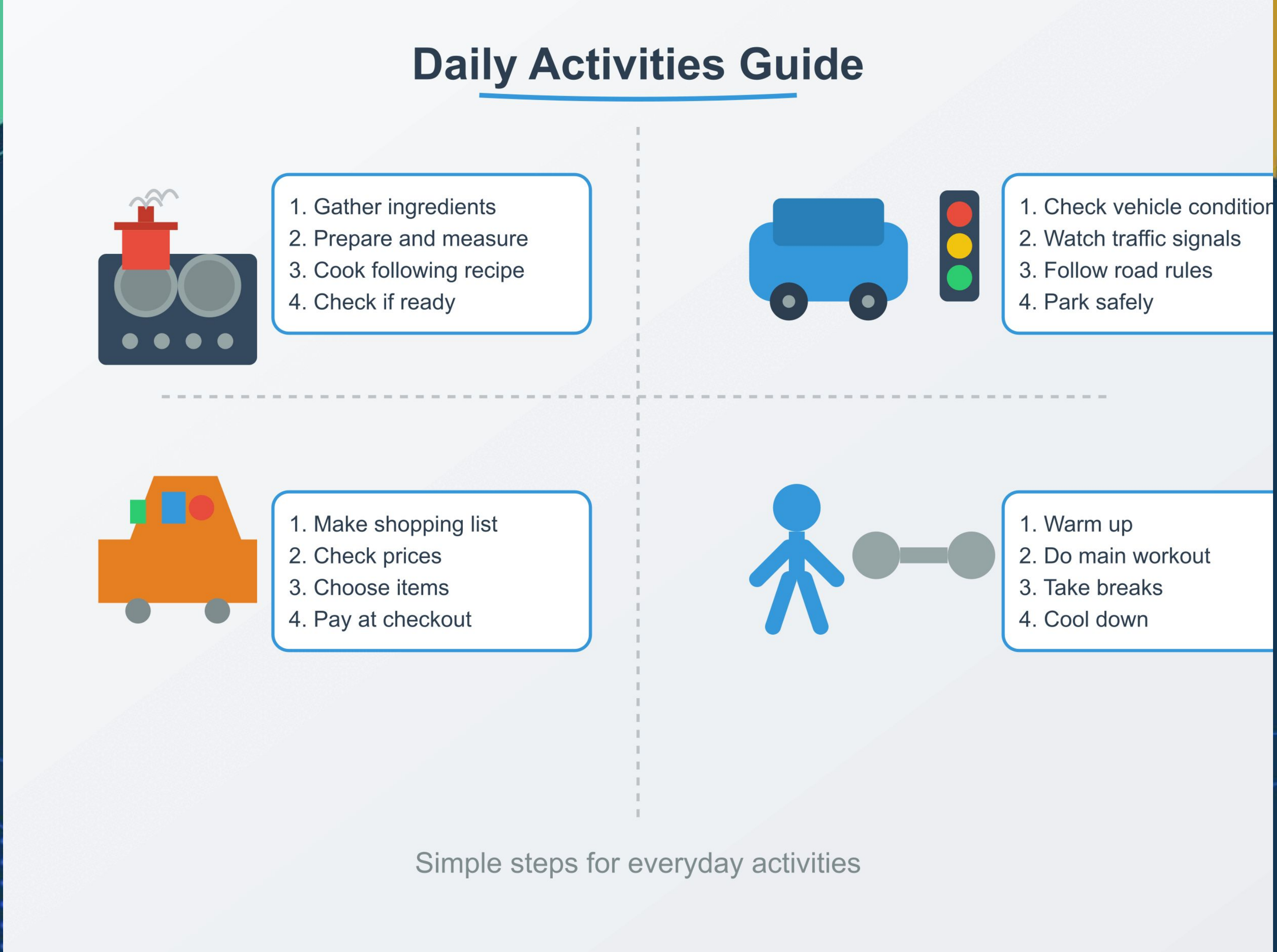
Copy link

via Francois Oberholzer Fwy Fastest route, the usual traffic Details	21 min 28.7 km
via Francois Oberholzer Fwy and Main Reef Rd/R29	30 min 26.6 km
via R24	25 min 25.7 km

Explore nearby O.R. Tambo International Airport

Introduction

CoGrammar



Daily Activities Guide

1. Gather ingredients
2. Prepare and measure
3. Cook following recipe
4. Check if ready

1. Check vehicle conditio
2. Watch traffic signals
3. Follow road rules
4. Park safely

1. Make shopping list
2. Check prices
3. Choose items
4. Pay at checkout

1. Warm up
2. Do main workout
3. Take breaks
4. Cool down

Simple steps for everyday activities

What is an Algorithm?

CoGrammar

Demystifying Algorithms

- **What is an Algorithm?**
  - An algorithm is a structured, step-by-step method for solving problems.
- **Why is it important?**
  - It provides a clear, logical approach to achieving desired outcomes.
- **Characteristics of a good Algorithm:**
  - Precision: the steps are precisely stated or defined.
  - Uniqueness: results of each step are uniquely defined and only depend on the input and the result of the preceding steps.
  - Finiteness: the algorithm always stops after a finite number of steps.
  - Input: the algorithm receives some input.
  - Output: the algorithm produces some output.

Why Algorithms Matter

- **Efficiency and Optimization:**
  - Algorithms help us find the most efficient solutions to problems.
  - By analyzing different algorithms, we can identify the best approach for a given task.
- **Problem-Solving and Logical Thinking:**
  - Algorithms encourage a structured and logical approach to problem-solving.
  - They help us break down complex problems into smaller, more manageable subproblems.
- **Characteristics of a good Algorithm:**
  - Algorithms are the building blocks of computer programs.
  - Understanding algorithms is essential for software development.
- **Real-world Applications:**
  - Search engines, recommendation systems, AI, and machine learning all rely on algorithms.

Solving Problems Like a Pro: Planning a Road Trip

- **Inputs**: Start point, destination, preferences
- **Process**: Route calculation steps
- **Outputs**: Route, time, stops
- **Success criteria**: Efficient & safe route

CoGrammar

Problem-Solving and Its Link to Algorithms

CoGrammar

Turning Problems into Solutions

- Analysing the problem:
    - Clearly define the problem and its goals.
    - Identify the inputs and desired outputs.
    - Consider any constraints or limitations.
- Break Down the Problem:
    - Divide the problem into smaller, more manageable subproblems.
    - This helps in simplifying the problem and focusing on individual parts.
- Plan the Solution:
    - Develop a step-by-step approach to solve each subproblem.
    - Consider different algorithms and data structures that can be used.
- Execute the Plan:
    - Implement the solution using a programming language or other tools.
    - Test the solution with various inputs to ensure correctness.
- Evaluate the Results:
    - Analyze the output of the algorithm to verify its accuracy.
    - Identify any errors or inefficiencies and make necessary improvements.

CoGrammar

Turning Problems into Solutions

- **Steps**:
  - **Analysing the problem:** Understand and analyze the problem clearly to identify inputs, outputs, and core functionalities.
  - **Plan the solution:** Devise and refine an algorithm to represent and select the best solution.
  - **Execute the plan:** Convert, execute, and document the solution.
  - **Reflect and optimize:** Test and validate the solution, optimizing and correcting errors as needed.

CoGrammar

Algorithm Design Techniques

- **Brute Force:**
  - Trying every possible solution until the correct one is found.
  - Often inefficient for large problem sizes.
- **Plan the solution:**
  - Breaking down a problem into smaller subproblems.
  - Solving each subproblem independently.
  - Combining the solutions to solve the original problem.
- **Greedy Algorithms:**
  - Making the best choice at each step, hoping to find the optimal solution.
  - May not always lead to the globally optimal solution.

CoGrammar

The Building Blocks of Algorithms

CoGrammar

The Core Ingredients of an Algorithm

- **Sequences**: Step-by-step instructions or Linear steps executed in order.
- **Conditions**: Decisions made based on logical checks
- **Loops**: Repeating actions until a condition is met.
- **Operations**: Processing data

CoGrammar

The Core Ingredients of an Algorithm: Sequences

- **<u>Algorithm for Making Tea:</u>**
  1. Boil water.
  2. Place a tea bag in a cup.
  3. Pour hot water into the cup.
  4. Wait for 3-5 minutes.
  5. Remove the tea bag and enjoy!

The Core Ingredients of an Algorithm: Conditions

- **Algorithm for turning on the lights**

```
1. Enter the room.
2. **If** the room is dark **then**:
      i. Turn on the lights.
3. **Else**:
      i. Keep the lights off.
4. Proceed with your activity in the room.
```

CoGrammar

The Core Ingredients of an Algorithm: Loops

- **Algorithm: Washing Dishes**

```
1. Collect all dirty dishes.
2. **While** there are dirty dishes:
    i. Pick up a dirty dish.
    ii. Wash and rinse the dish.
    iii. Place it in the drying rack.
3. **Stop** when all dishes are clean.
```

CoGrammar

The Core Ingredients of an Algorithm: Loops

- **Algorithm: Assigning Chores to Family Members**

```
1. Make a list of family members: [Alice, Bob, Charlie].
2. Make a list of chores: [Dishes, Vacuuming, Trash].
3. For each family member in the list:
    i. Assign the next chore from the list.
4. Stop once all family members have a chore.
```

CoGrammar

The Pseudocode

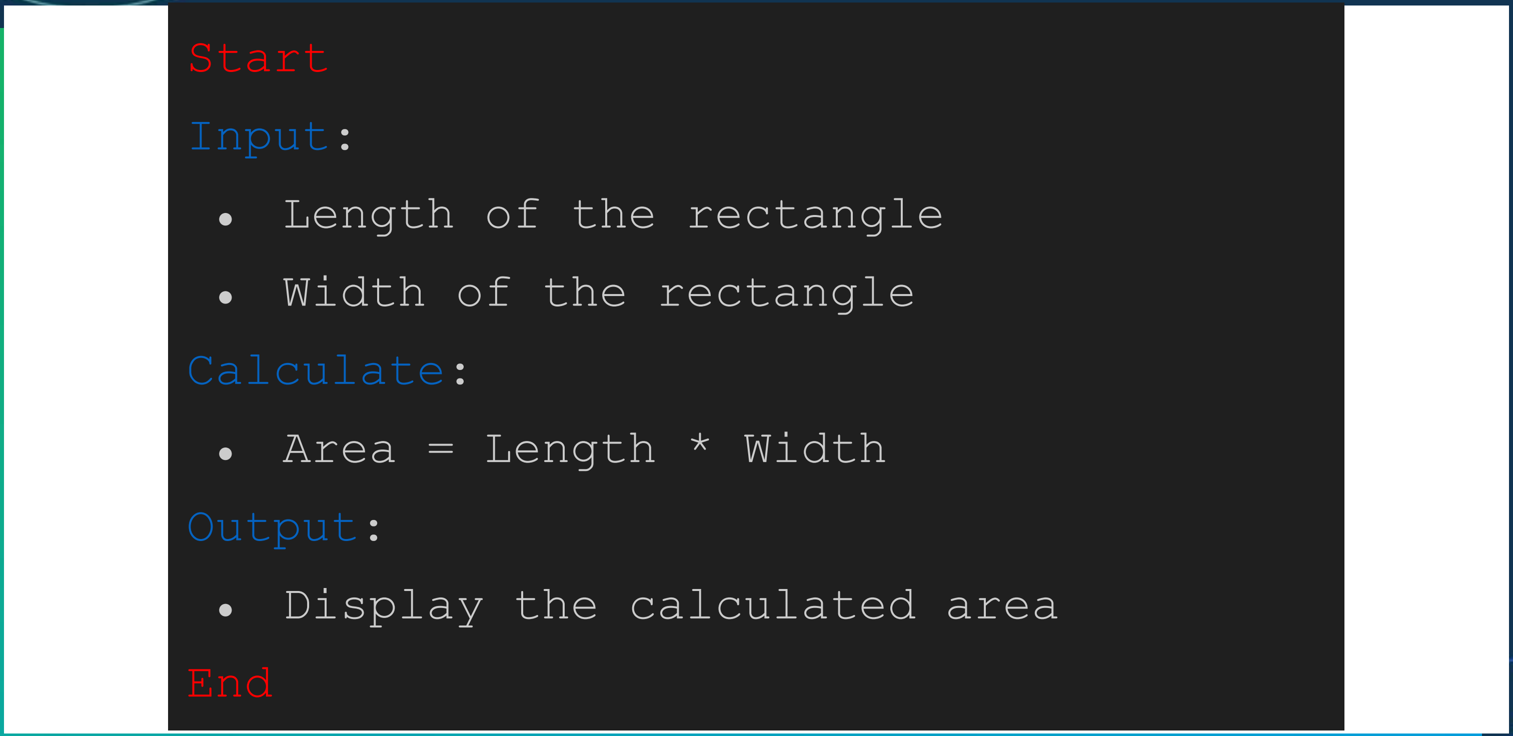
CoGrammar

Understanding Pseudocode

- What is Pseudocode?
    - A simplified programming language that uses plain language to describe the logic of an algorithm.
    - Serves as a bridge between **human thinking** and **machine implementation**.
    - It's like a blueprint for your program.

- Why Use Pseudocode?
    - **Clarity and Conciseness:** It provides a clear and concise representation of the algorithm.
    - **Language Independence:** It's not tied to a specific programming language, making it versatile.
    - **Problem-Solving Tool:** It helps in breaking down complex problems into smaller, manageable steps.

Understanding Pseudocode



```
Start
Input:
    •  Length of the rectangle
    •  Width of the rectangle
Calculate:
    •  Area = Length * Width
Output:
    •  Display the calculated area
End
```

CoGrammar

Let’s build! The Odd/Even Checker

- Display or print numbers from 1 to 10
- Identify odd/even using conditions and loops

Practical: Pathfinder Algorithm for Navigating a Maze

CoGrammar

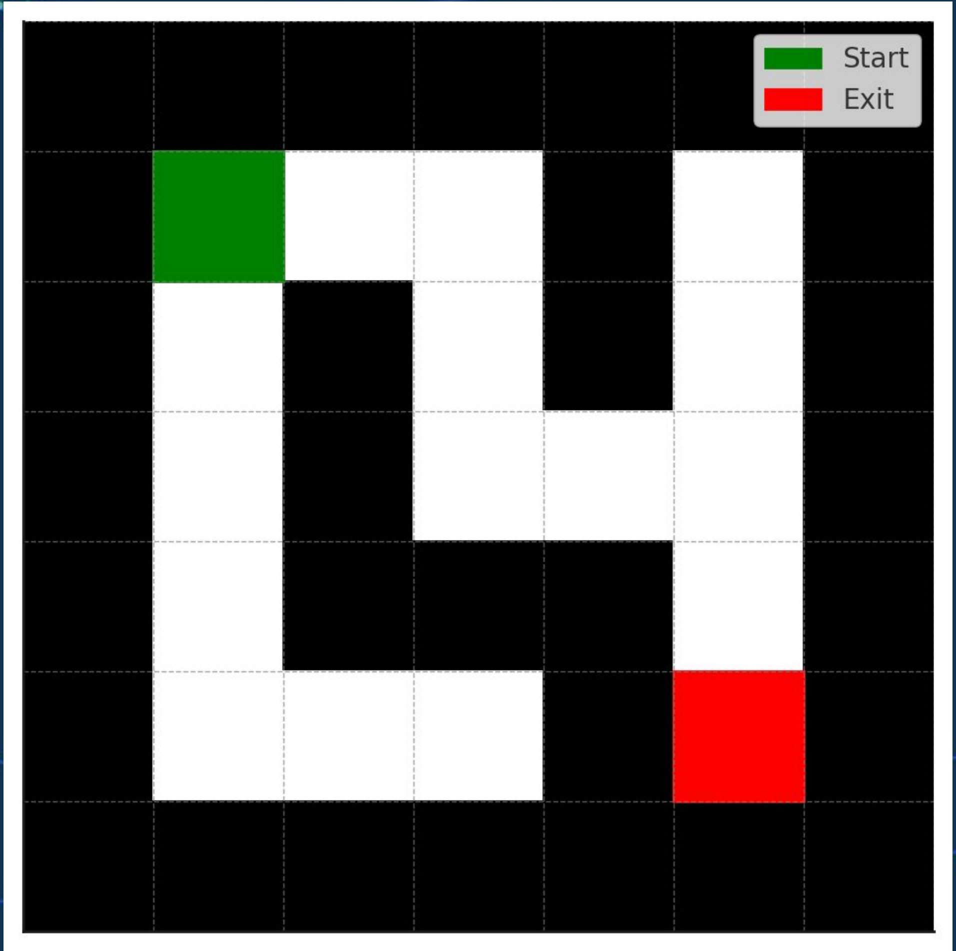
Finding the Way Out: Maze Navigation

- **Task:** Navigate a maze using the left-hand rule.
- **Concept:** Always turn left if possible; move straight otherwise; turn right only as a last resort.

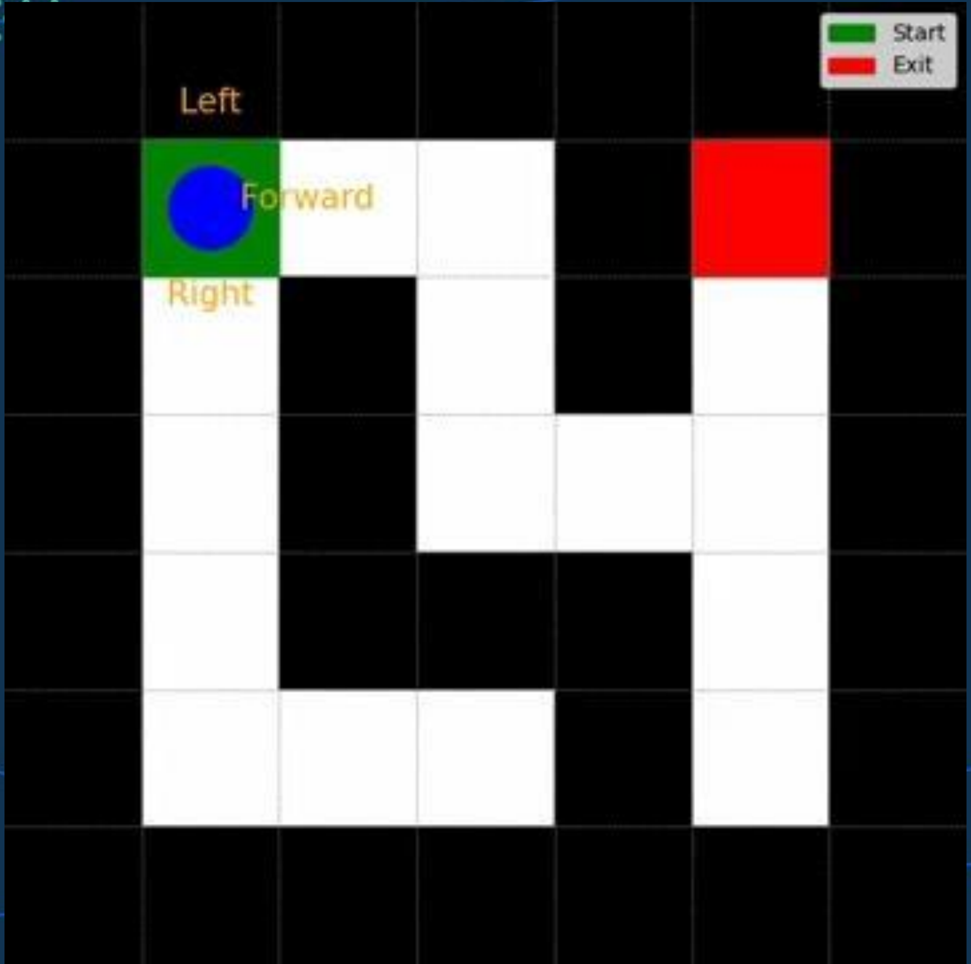
```
Start at the maze entrance.
while not at the exit:
    if there is an open path to the left then:
        Turn left, mark as passed and move forward.
    else if there is an open path straight ahead then:
        Mark as passed and move forward.
    else if there is an open path to the right then:
        Turn right, mark as passed and move forward.
    else:
        Turn around (backtrack).
print "Maze solved. Exit reached.
```

CoGrammar

Finding the Way Out: Maze Navigation



Finding the Way Out: Maze Navigation



CoGrammar

Poll

A bank uses an algorithm to approve transactions based on predefined rules. For example, transactions over $10,000 require manager approval, while transactions below $10,000 are processed automatically. Which steps reflect an algorithmic approach for this process?

1. Approve all transactions automatically to speed up processing.
2. Check the transaction amount; if greater than $10,000, send it for manager approval.
3. Check the transaction amount; if less than or equal to $10,000, process it automatically.

CoGrammar

Poll

Uber assigns drivers to riders based on multiple factors. Which examples describe efficient algorithms for this process?

1. Randomly assigning drivers without considering distance.
2. Matching the nearest available driver to the rider based on location.
3. Balancing workload by assigning a driver with the fewest trips completed that day.

CoGrammar

Lesson Conclusion and Recap

Recap the key concepts and techniques covered during the lesson.

- **What is an Algorithm?**: Algorithms are structured, step-by-step methods to solve problems. They help translate complex processes into logical sequences that computers can execute.
- **Problem-Solving and Algorithms**: Problem-solving involves four steps: Understand, Plan, Execute, and Reflect. Algorithms formalise this process by defining inputs, outputs, and logical steps.
- **The Three Types of Instructions**: **Sequences**: Linear, step-by-step execution. **Conditionals**: Decision-making based on conditions (e.g., if-else). **Loops**: Repeating steps until a condition is met.
- **Applying Concepts to Real-World Problems**: Algorithms like navigation systems (e.g., road trips, maze solving) demonstrate how these ideas are used in practice. Emphasis on logical steps and ensuring repeatability and efficiency..

CoGrammar

Resources

Resources

- **Web Pages:**
  - [Scratch Practice]
  - [Make a Flappy game]

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar