WHAT IS A PROBLEM STATEMENT?

A problem statement is a clear description of a challenge or goal that learners need to solve. It presents a situation that requires computational thinking and/or programming to resolve.

A problem statement tells learners what needs to happen, but not exactly how to make it happen. Learners must figure out the solution themselves.

When problem statements are well-crafted, learners stop seeing lessons as tasks to complete and start seeing themselves as capable problem-solvers who can help others - a powerful foundation for lifelong learning in computer science and beyond.

In Coding and Robotics, we solve problems!

The Structure of a Good Problem Statement - Essential Elements:

- **A Clear Goal/Challenge**
  What needs to be achieved?
- **Context or Scenario** (when appropriate)
  Why does this matter? Who needs this solution?
- **Constraints or Rules**
  What are the limitations or requirements?
- **Success Criteria** (sometimes implicit)
  How will we know the problem is solved?

WHY USE PROBLEM STATEMENTS?

1. Develops Computational Thinking
Problem statements require learners to:

- **Decompose** (break the problem into smaller steps)
- **Recognise patterns** (see **what** repeats or is similar)
- **Abstract** (focus on important information)
- **Create algorithms** (design step-**by**-step solutions)
- **Debug** (test and fix when it doesn't work)

2. Promotes Active Learning
When faced with a problem, learners must:

- Think **critically**
- Make **decisions**
- Try different **approaches**
- Learn from **mistakes**

This is much more powerful than passively following instructions.

3. Builds Problem-Solving Skills
Real programmers and engineers don't get told exactly what to do - they're given problems to solve. Problem statements prepare learners for authentic problem-solving.

4. Encourages Multiple Solutions
Good problem statements often have more than one correct solution. This:

- Values different **thinking** approaches

- Promotes **creativity**
- Enables **differentiation** (some solutions are simpler, others more complex)
- Creates opportunities for rich discussion: "How did you solve it? How is yours different from mine?"

5. Makes Learning Meaningful
Problems give **context** and **purpose** to coding and robotics. Instead of learning "what is a loop," learners think "I need to make this action repeat - how can I do that?"

6. Aligns with Real-World Application
In real life, we don't follow recipes - we solve problems. Problem statements mirror how coding and robotics are actually used in the world.

WHAT MAKES A GOOD PROBLEM STATEMENT FOR FOUNDATION PHASE?

A high-quality problem statement for FP learners should:

- **Be age-appropriate**: Use simple language and familiar contexts
- **Create empathy**: Feature relatable characters or situations
- **Be specific**: Clearly define what needs to be accomplished
- **Motivate learning**: Create genuine interest in solving the problem
- **Connect to real life**: Use scenarios children can relate to
- **Support learning intentions**: Directly address the competencies being taught
- **Be scaffolded**: Provide appropriate challenge without overwhelming
- **Encourage agency**: Make learners feel they can help/solve the problem

PEDAGOGICAL OPPORTUNITY

In Grade 3, problem statements in lesson plans can occasionally be used as an opportunity to develop aspects of computational thinking, such as abstraction and even pattern recognition.

For example (*abstraction*), when a problem statement has unnecessary detail, then guide them to identify what is essential. Use these opportunities to reinforce computational thinking (*abstraction* and *pattern recognition*) as described earlier.

Strategies
For Grade 3, teach learners to ask three questions about any problem:

1. **Use the "What, Where, How" Framework**

Grade 3 Example: *"There are toys scattered all over the classroom floor after playtime. Some are cars, some are blocks, and some are puzzles. We need the robot to collect all the toys and put them in the toy box in the corner."*

Guide learners to extract:

- **What** needs to happen? → Collect toys
- **Where** does it happen? → From floor to toy box
- **How** will we do it? → Robot picks up and moves items

This reduces the problem to its computational essentials.

2. **The "Important Information Hunt"**

Make it a game to find necessary vs. unnecessary information.

Grade 3 Activity: Give problem statements with a mix of relevant and irrelevant details.

"It's raining outside. Thabo's robot must turn left at the red block, then turn right at the green block. Thabo is wearing his blue shirt today. The robot must then move forward 2 steps."
Learners:

- ✓ Mark important information (with green)
- ✗ Mark extra information (with red)
- Rewrite using only green-marked information

3. The "Story vs. Code" Comparison

Show learners how we translate rich contexts into simple instructions.

Grade 2/3 Process:

1. Read a story-based problem together
2. Ask: "If we told this to a robot, what would we say?"
3. Create two columns on the board:

Story Version	Robot Instructions
"The hungry caterpillar wants to eat the apple"	Move to apple, eat
"It's a red, juicy apple on a green leaf"	(not needed for coding!)

Grade 3 Progression:

- Start where Grade 2 ended, then add the following:
- Add: "Why did you keep this information? Why did you remove that?"
- Add: Learners create their own simplified version and compare with peers

For Teachers – Implementation Tip: Do not do this with EVERY problem statement - it becomes tedious. Instead:

- **Weeks 1-3:** Explicitly teach abstraction using 1-2 problem statements per week
- **Weeks 4-10:** Do quick checks occasionally ("What's this problem really asking?")
- **Term 2+:** Learners should be identifying main ideas more independently
- The goal is to make this thinking habitual rather than constantly explicit.

For Assessment:
Use problem statements to assess:

- Can learners identify what the problem is?
- Can they suggest appropriate solutions?
- Do they connect a problem to learned concepts?
- Can they explain their problem-solving process?

Remember
Effective problem statements are the heart of inquiry-based learning. By improving problem statements to be more engaging, specific, and aligned with learning objectives, we create authentic reasons for Foundation Phase learners to engage with coding and robotics concepts.

CONCLUSION:

Problem Statements

- Create emotional investment through relatable characters
- Provide clear, achievable goals
- Embed computational thinking language naturally
- Connect practical activities to theoretical concepts
- Respect learner agency and capability
- Build progressively in complexity
- Reflect real-world applications