# 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!

**1 The Structure of a Good Problem Statement - Essential Elements:**

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

Also refer to section on:

***Learner Readable Problem Statements for Grade 1*** (p2-4) for pedagogical opportunity.

## WHY USE PROBLEM STATEMENTS?

**1. Develops Computational Thinking**
Problem statements require learners to:

- **Decompose** (break the problem into smaller steps)
- **Recognize 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 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 objectives**: Directly address the competencies being taught
- **Be scaffolded**: Provide appropriate challenge without overwhelming
- **Encourage agency**: Make learners feel they can help/solve the problem

## LEARNER READABLE PROBLEM STATEMENTS FOR GRADE 1

### Grade 1 Term 1 Coding & Robotics

The following problem statements are simple versions that learners can read independently.

These problem statements are written at a Grade 1 reading level so learners can read them independently or with minimal support. Each problem uses:

- **Simple**, high-frequency words (can, help, the, go, etc.)
- **Short** sentences (5-10 words maximum)
- **Decodable** words at Grade 1 level
- **Clear**, simple punctuation
- Century Gothic font (14-16pt) for better readability (increase font size in examples below)

### About These Problem Statements

### How to Use

- **Display:** Write on board or display on screen in large font
- **Read Together:** First read aloud together as a class
- **Independent Reading:** Have learners read individually or in pairs

- **Support:** Use pictures/visuals alongside text
- **Teacher Version:** You can still use the longer teacher version to introduce the lesson, then show the simple version

**Learner Readable Problem Statements for Display**

*These can be printed or displayed for learners to read*

**WEEK 1 - ME**

**Help Bogosi!**

**Bogosi is lost.**
**He cannot find his line.**
**Can you help him?**

**Use the arrows.**

**WEEK 2 - ME**

**Look at the game!**

**Patty says triangle.**
**Hlengiwe sits down.**
**Patty says circle.**
**Hlengiwe jumps up.**

**Is this a pattern?**

**WEEK 3 - MY SCHOOL**

**Help Pulane!**

**Pulane is new.**
**She cannot find the toilet.**
**Can you help her?**

**Make a code to show the way.**

**WEEK 4 - MY SCHOOL**

**Let us build!**

**We need a new classroom.**
**Look at the shapes.**
**What shapes do you see?**

**Let us build it together!**

**WEEK 5 - MY SCHOOL**
Find the school!

Biancke is lost.
Her friend's school is red.
It has a yellow roof.

Can you find it?

**WEEK 6 - HEALTHY LIVING**
Help Grace!

Grace needs oranges.
Her basket is empty.
Can you help her?

Make a code to get the oranges.

**WEEK 7 - HEALTHY LIVING**
Pack a lunch!

Dzunani needs lunch.
He wants 4 healthy foods.

Can you help him pick?

**WEEK 8 - MY FAMILY**
Make a frame!

Tshepo has a photo.
He needs a frame.

Follow the steps.
Work with a friend!

**WEEK 9 - THE WEATHER**
Look at the pattern!

Palesa sees a pattern.
It is on the window.

Can you see it too?
What comes next?

**WEEK 10 - WEATHER**
**It is hot!**

**Elizabeth is hot.**
**She needs shade.**
**The tree has shade.**

**Make a code to help her go!**

**Note: Pedagogical Opportunity**

The simplified problem statements could also be used as excellent teaching tools *if* you occasionally show learners how they have been simplified through using abstraction (identifying essential information and removing unnecessary details), making the abstraction thinking visible to model the skill by doing the following.

- **Make the process visible** - Show them both versions (complex and simple) and discuss what was kept and what was removed, and why.
- **Guide them through simplification** - Give them practice identifying "what matters most" in a problem or situation.
- **Use concrete examples** - "This long story has lots of details, but the important parts for solving our problem are just these three things..."
- **Build metacognitive awareness** - Help them understand that they are learning to "find the important parts" or "make things simpler"

**For Grade 1, developmentally appropriate abstraction might look like:**

- Identifying the main character and goal in a story problem
- Recognising what information they need vs. do not need
- Simplifying a multi-step real-world scenario to its key actions
- Finding patterns by noticing what's the same and ignoring what's different

**Recommendations for Implementing Problem Statements**

**Introduce Problems Dramatically**

- Use props, voices, or puppets when presenting problem statements
- Build anticipation: "Oh no! We have a problem today!"
- Make it feel urgent but not stressful

**Keep Characters Consistent When Possible**

- Consider having 2-3 recurring characters throughout term
- Examples: "Robot Rosie" could appear in multiple weeks, "Boppy the Robot" could return
- Builds familiarity and investment

**Allow Learner Input**

- After presenting problem, ask: "What do you think we should do first?"
- Let learners suggest solutions before diving into activity
- Validate their thinking even if guiding toward intended approach

**For Teachers:**

- **Read problem statement with enthusiasm** - Your energy matters!
- **Use different voices for different characters**
- **Pause for student reactions** - Let them process the problem
- **Scaffold as needed** - Some students may need problems broken down
- **Celebrate all attempts -** Focus on problem-solving process, not just answers
- **Use the pedagogical opportunity –** Use the opportunity to reinforce abstraction as described earlier.

**For Assessment:**

Use problem statements to assess:

- Can learners identify what the problem is?
- Can they suggest appropriate solutions?
- Do they connect 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.

**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