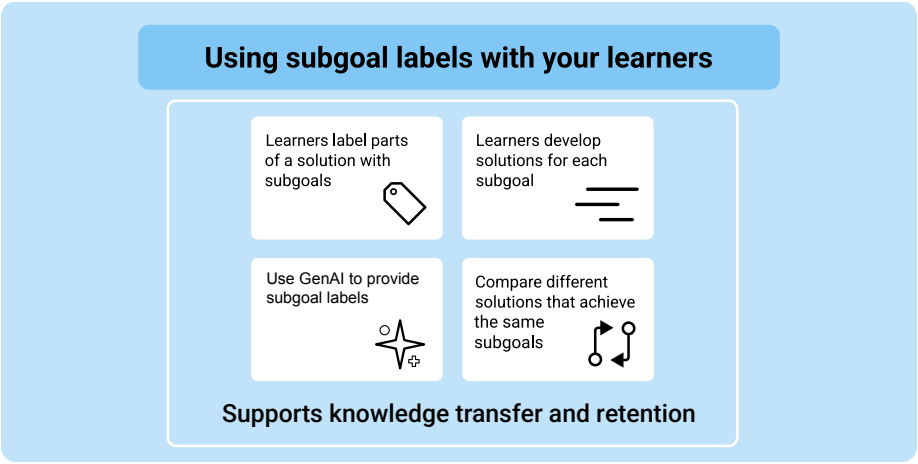
Raspberry Pi Foundation

Pedagogy Quick Read

Subgoal labels help novice programmers persist and improve problem solving

Subgoal labels break down problems into pieces small enough for novices to grasp. They are an instructional design framework that can be embedded into product-oriented worked examples to improve learner performance during problem solving. They have been shown to be effective in many disciplines, including computing.[1] Learner performance improves because subgoal labels help learners focus on the problem-solving process while increasing the transfer and retention of knowledge.



What are subgoal labels?

Subgoal labels explicitly teach learners the functional pieces of a problem-solving procedure. They are an instructional tool that is designed to bridge the gap between novices and experts (learners and instructors). Learners usually encounter them through worked examples that have some or all subgoals labelled. While the specific steps needed to achieve a subgoal varies from problem to problem, the underlying problem-solving pieces remain the same.

Subgoal labels help educators:

- Explicitly teach the functional pieces (subgoals) of a problem-solving procedure
- Make their internalised tacit knowledge explicit to learners

Subgoal labels help learners:

- Focus on the problem-solving process rather than superficial details
- Gain a deeper understanding of the material
- Retain knowledge longer and solve novel problems more accurately
- Persist and succeed in introductory programming courses

Subgoal labels are best when used to:

- Generalise problem-solving steps that can be transferred to other problems, rather than being problem-specific
- Introduce a new concept, and can be revisited when the concept re-occurs
- Illustrate the shared functional features in worked examples, which helps learners to organise information so that they can transfer it more easily

Subgoal labels reduce cognitive load

Worked examples can be a good way to introduce new programming topics. However, they can also add cognitive load because worked examples often refer to a specific context, and learners must process that extraneous contextual information.[2] Adding subgoal labels to worked examples helps learners focus on the steps needed to solve the problem, rather than the context of the problem, reducing the cognitive load.[3] Subgoal labels help learners organise information and focus on the structural features of the worked example.

Subgoal labels help learners transfer knowledge

Novices may struggle in solving introductory programming problems. And once they have solved the problem, they often do not recognise which pieces of the solution can be transferred to solve other problems. This transfer of knowledge is critical when learning programming as each new problem likely has a different context. Learning with subgoal labels addresses the problem of transfer by pointing out the common patterns and procedural problem-solving steps so that learners can apply them across different problems. Research indicates that learning with subgoal labels helps learners solve novel problems more accurately and retain knowledge longer.[1,4,5]

Subgoal labels help make tacit knowledge explicit

Computing educators are experts in the programming process — they have internalised and automated much of the low-level knowledge and basic programming skills that students are just starting to learn. Because experts can make decisions about programming without conscious processing, it can be very difficult for teachers to explain their processes and decisions to novices.

To the expert, these decisions feel like common knowledge, intuition, or "just the way you do it". For novices, these decisions are not automatic and require thought and practice. Continued practice and repetition help make them automatic for the learner.

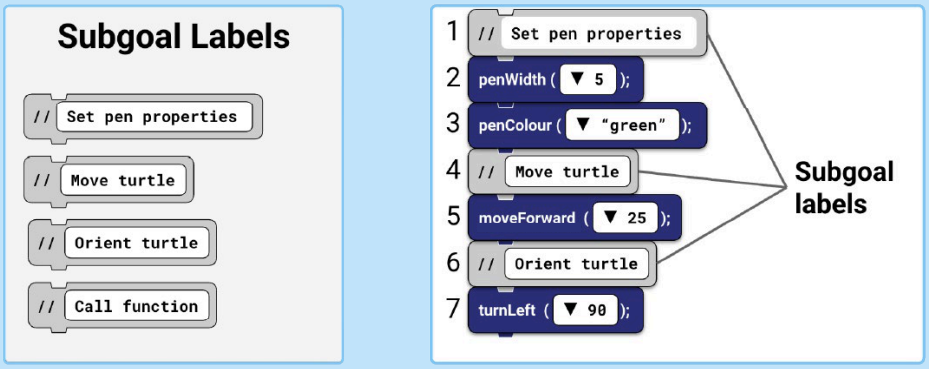
Using subgoal labels while teaching programming uncovers and communicates both the tacit and conceptual knowledge that the expert has internalised. For the learner, the subgoal labels explicitly provide the scaffolds needed to help them internalise this knowledge and become experts themselves.

Subgoal labels help learners persist and perform better

Novice programmers who learnt using subgoal labels demonstrated a deeper and more complex understanding of the materials:

- In lab-based studies using a block-based programming language, learners performed 7–8% better on assessments when they received subgoal-oriented learning materials[5]
- In a Java-based introductory programming course, subgoal-oriented materials helped students perform better, especially those who were struggling6

Encouragingly, students in the subgoal group were half as likely to withdraw and half as likely to fail compared to the control group,[1] learning using subgoal labelled worked examples helps learners persist with programming.



How to use subgoal labels in your classroom

There are several ways to use subgoal labels with your learners:

- You can provide subgoal labels that have been created by an expert
- You, or your students, can ask an AI tool to provide subgoal labels[7]
- You can teach students how to write their own subgoal labels

You will need to provide guidance to ensure that the labels are general enough to work across problems and illustrate to your learners that different problems can be solved with the same subgoal labels.

To learn more and access resources, visit cs1subgoals.org, watch this webinar, or check out our Research Seminar on the topic.

References

[1] Margulieux, L.E. et al. (2020). Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. the-cc.io/qr31_1

[2] Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. the-cc.io/qr31_2

[3] Atkinson, R.K. et al. (2000). Learning from examples: Instructional principles from the worked examples research. the-cc.io/qr31_3

4Catrambone, R. (1998). The subgoal learning model: Creating better examples so that students can solve novel problems. the-cc.io/qr31_4

5Margulieux, L.E. et al. (2012). Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. the-cc.io/qr31_5

6Sweller, J. (2010a). Element interactivity and intrinsic, extraneous, and germane cognitive load. the-cc.io/qr31_6

7Marwan, S. et al. (2025). How good are large language models at generating subgoal labels? the-cc.io/qr31_7

Raspberry Pi Foundation