Barefoot
from Computing at School | BT

What is PRIMM?

PRIMM is a simple way to create sets of carefully designed lesson activities to help teach programming. The theory was developed by a computing educational researcher, Sue Sentance, to help teach Python, a text- based programming language, in secondary schools in England. In Sue’s research, PRIMM was seen to improve pupil’s learning [1].

We do not yet have evidence that PRIMM works for other programming languages, such as block-based programming languages like Scratch. However, we think that the approach could be just as effective, as the research on which PRIMM builds is very well regarded.

In this short article, we introduce you to PRIMM, and we hope you might have a go at using some of the ideas in your own teaching. This could be when using BeeBots, Scratch, Logo, Crumbles or the MicroBit. PRIMM is not just for Python!

1) What is PRIMM?

PRIMM stands for 5 stages in teaching programming, of **Predict**, **Run**, **Investigate**, **Modify** and **Make**. In PRIMM, these stages are taught in sequence. Each of these stages is based on underpinning research. Some of this research is from university studies, some from teaching younger learners, but all of it is very well-regarded work. For example, the predict and run stages build on research that shows that novice programmers need to learn to read code before they can write it.

2) What are the stages of PRIMM?

The stages are:

Predict	Learners are shown a sample program, they independently or in pairs, read the code and predict what it will do when it is run. The learners are asked to record their prediction, e.g. by writing it down or drawing what they think will happen.
Run	Learners are given the sample program to run. They run the program and see if their prediction was correct. Often at this point, there are discussions about what surprised them or what happened as expected.
Investigate	Learners are given a set of questions or short tasks to help them investigate important ideas in the program. Learners complete the tasks and record their findings. The investigate tasks might be for learners to add comments to the code, to explore aspects of the code, and even to make specific changes to see what happens.
Modify	Learners are asked to take the program and make it their own, they are asked to change it to do something different. For example, they could change the context and questions for a quiz, from say a history quiz to a geography quiz.
Make	Learners use what they have learned to make something new. For example, they might have learned about using if statements (selection) in quiz context and then use this to make an interactive animation.

3) How long does *a PRIMM sequence* take?

In the secondary PRIMM research, some PRIMM sequences took a long 90-minute lesson, others took three or more hour-long lessons. This was dependent on the teacher who delivered the lessons, what they focused on and on the learners themselves. We have created two example Barefoot PRIMM lesson plans based on the original *Crystal Flowers* resource. You can use these to show you how to integrate PRIMM into your teaching.

4) Where do the sample programs come from?

The teacher who creates a PRIMM sequence creates the sample program. The teacher chooses a concept (or set of concepts) that they want pupils to learn about and creates a well-written example that showcases these. For example, if you wanted your pupils to learn about *if* statements (selection) and asking and answering questions, you might choose a quiz or an interactive animation as your context. Suppose you wanted your pupils to learn about *repetition*; in that case, you might create a sample program that draws shapes or that animates a character performing a PE routine.

5) Where do the investigate questions and tasks come from?

The teacher who creates the PRIMM sequence creates the investigate tasks. These tasks might include a question about the type of programming construct that is being used and ask pupils to spot it in the program. For example, find the *if* statement (selection) and highlight it (if the code is printed) or add a comment to show where it is in an online example. There might be a task to change a significant aspect of a command in the program. For example, in a repeat loop, the pupils might be asked to change the number of times the loop runs and to then see what the effect is.

6) What about modify and make stages? How do you decide what to ask pupils to do for these?

As with the earlier stages, the teacher who creates the PRIMM activity thinks what will help their learners to gain a deeper understanding of the concepts they are teaching. For example, in a modify activity where students are learning about selection statements, the teacher might ask students to change this aspect, by introducing a new context. The students can keep the general structure in the sample program but need to change the details. In the make, the teacher is looking for another context in which the students can apply their new skills. Sometimes the **Modify** of PRIMM develops into a longer **Make**. But we need to do more research to look at the impact of this.

To try a PRIMM lesson sequence, try the Barefoot PRIMM *Crystal Flowers* activity; **PRIMM 1**, **PRIMM 2** which you will find under lessons in the resources section of the website. Barefoot have converted the original *Crystal Flowers* lesson to follow the PRIMM stages. More research is needed to study PRIMM in primary, but we think there is much to gain from trying it out with your class.

If you would like to find out more about PRIMM, look at Sue Sentences blog on PRIMM [2] or at her website on PRIMM [3] or read one of the research papers on PRIMM [1].

References

1 *Teachers' Experiences of using* PRIMM *to Teach Programming in School, 2019, Sentence S., Waite J., and Kalia M. SIGCSE '19: Proceedings of the 50th ACM Technical Symposium on Computer Science Education February 2019 Pages 476–482 https://doi.org/10.1145/3287324.3287477 Pre-print available https://primming.files.wordpress.com/2019/01/pre_print_teachers__experiences_of_using_primm_to_teach_programming_in_school.pdf*

2 PRIMM*: a structured approach to teaching programming, 2017 King's College London Computer Science Education Blog,*
https://blogs.kcl.ac.uk/cser/2017/09/01/primm-a-structured-approach-to-teaching-programming/

3 *Sue Sentance's Website with pages on* PRIMM
https://suesentance.net/2019/05/06/new-paper-primm-and-sociocultural-theory/