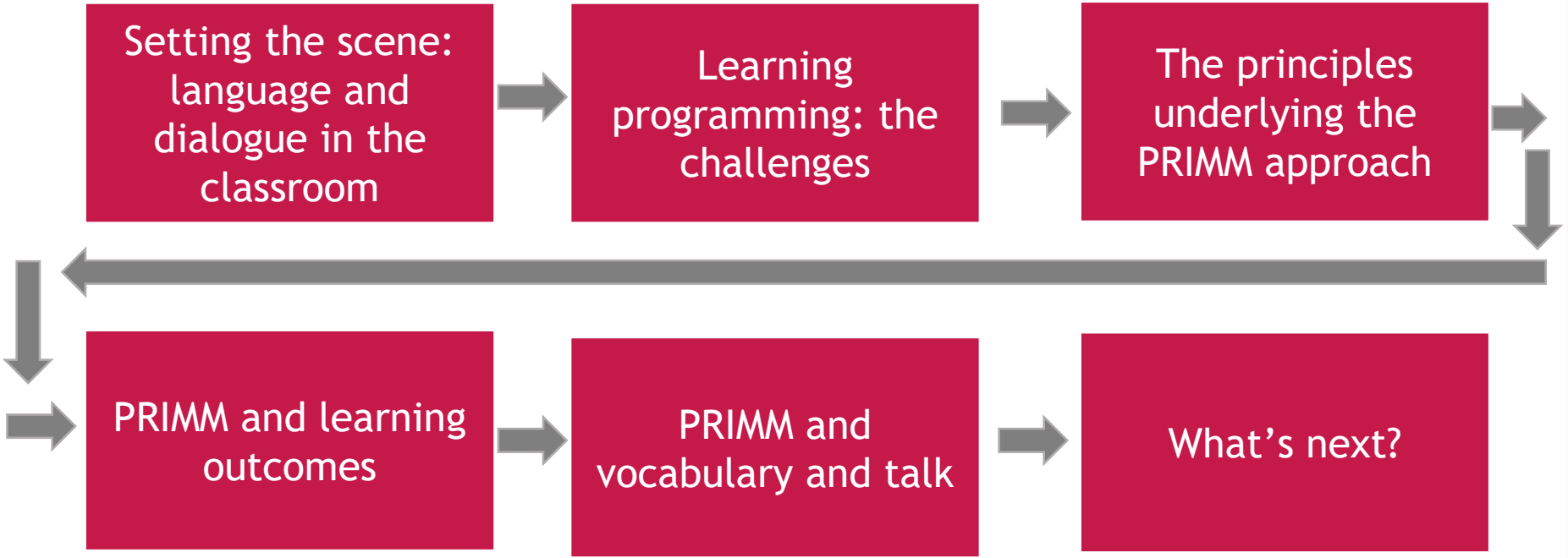
PRIMM:
its role in dialogue and vocabulary development in programming lessons

Dr Sue Sentance

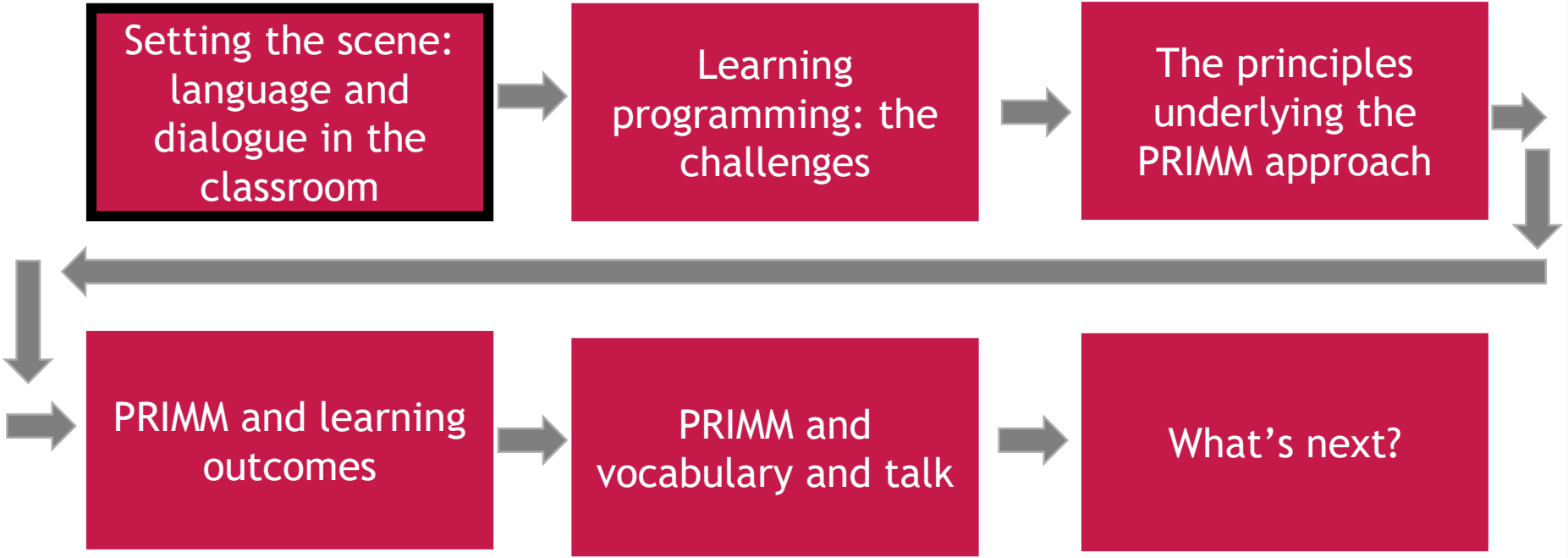
3rd November 2020

@suesentance

Overview of seminar



Overview of seminar



teacher

students

explaining

instructing

questioning

listening

modelling

sharing

clarifying

discussing

guiding

students

explaining

modelling

clarifying

discussing

problem-solving

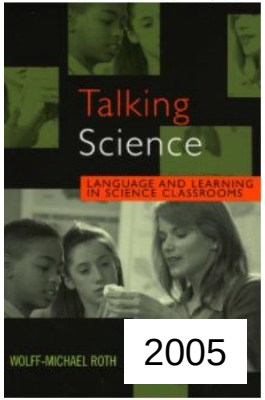
questioning

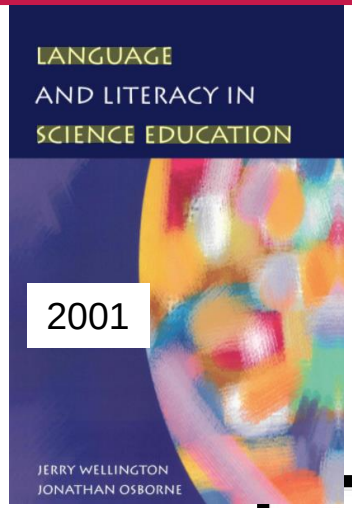
listening

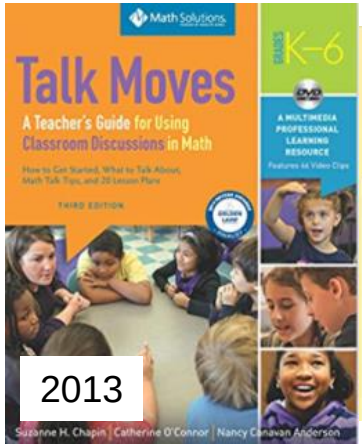
reasoning

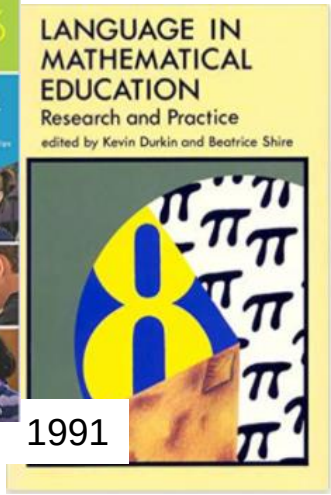
sharing

Language in education



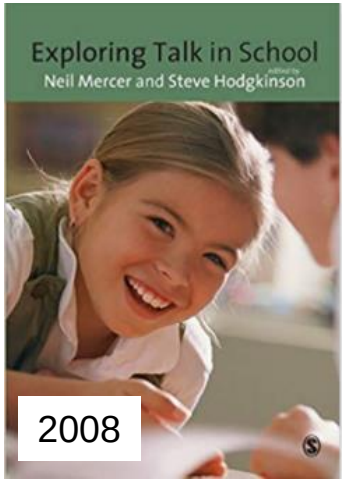


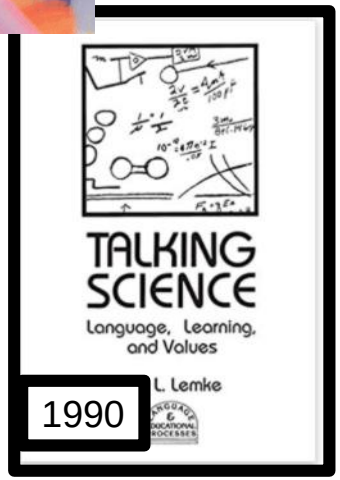




There is a LOT of research and guidance around language and education

… but not really in computing education yet (we are a young subject)





Language in education research

- Dialogic education
  - Dialogically organised instruction
  - ‘Thinking Together’
  - Dialogue and the impact on learning outcomes
- Vocabulary
  - Key terms
  - Strategies to learn vocabulary

Dialogically organised instruction

Nystrand et al (2003) described three teacher discourse moves frequently used by dialogic teachers to organise instruction in a coherent manner:

- **Uptake** : teacher validates particular students' ideas by incorporating their responses into subsequent questions thereby building upon and extending students' contributions
- **Authentic questions:** used to explore students' views and ideas rather than test their knowledge; reflect teachers' genuine interest in interacting with students, which helps to create a culture of dialogic education.
- **High-level evaluation :** consists of both the teacher's certification of the students' response and incorporation of the response into the discourse of the class, usually in the form of either an elaboration or a follow-up question'

Thinking Together – Mercer & colleagues

This research provides evidence for a sociocultural (Vygotskian) view of learning where social interaction has a developmental influence on individual thinking and intellectual outcomes

Thinking together was a longitudinal research project demonstrating the impact of a dialogue-based approach on children's learning and thinking.

Mercer & Sams (2006):

- providing children with guidance and practice in how to use language for reasoning enables them to use language more effectively as a tool for working on maths problems together.
- improving the quality of children's use of language for reasoning together improves their individual learning and understanding of mathematics.
- the teacher is an important model and guide for pupils' use of language for reasoning.

Neil Mercer & Claire Sams (2006) Teaching Children How to Use Language to Solve Maths Problems, Language and Education, 20:6, 507-528,

Dialogue – leading to learning outcomes

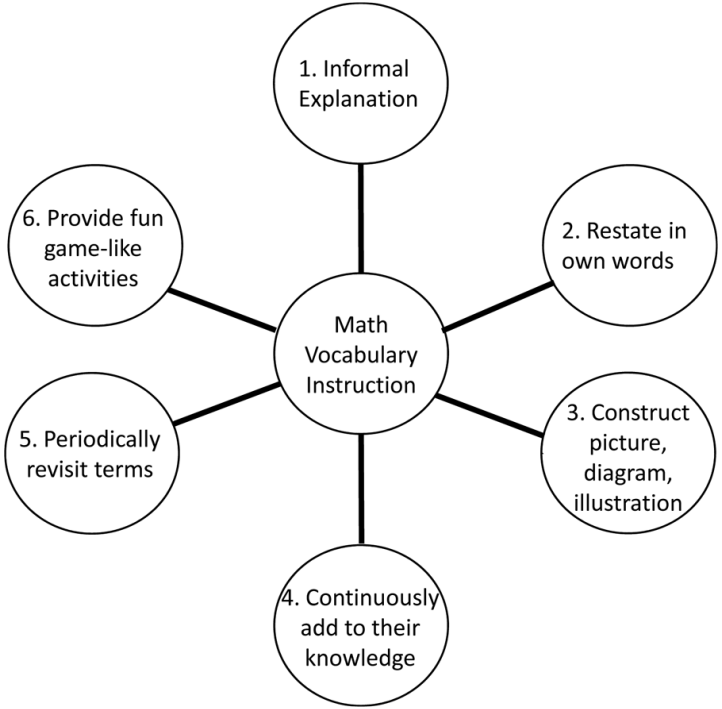
The Cambridge Classroom Dialogue Project (2015-2018)

In maths, literacy and science classrooms, Howe et al (2019) found three aspects of teacher-student dialogue strongly predicted performance in assessments:

1) **Elaboration** - where building on/elaborating /evaluating/clarifying of a previous contribution was invited or provided
2) **Querying** - where a previous contribution was doubted/challenged/rejected
3) **Student Participation** - where across the lesson multiple students were seen to engage with each other's ideas, and not merely respond to their teacher's questions.

What about learning the key terms and vocabulary in a subject?

- Teachers need to use research-validated instructional methods to teach important vocabulary [in mathematics]" (Ricommini et al, 2015)
- Academic language support is important for all learners – ongoing development of explicit [maths] vocabulary is essential (Bay-Williams & Livers, 2009)
- Engage students in "meta discourse" – talk to them about the informal everyday language around computers and compare with scientific language (Diethelm, Goschler & Lampe, 2018)



Key words in school programming education

Iteration
Array
Dictionary
Sequence
File
Selection
True
Abstraction
Sub-routine

Variable
Output
Function
Parameter
Argument
Constant
List
Input
Print

Algorithm
Data structure
Record
String
Integer
Search
Sort
Boolean

False
Condition
Program
Procedure
Bug
Object
Code
Editor
Data type

If … then
For loop
Source
While

Which of these terms should learners be able to use in explaining their programs at a) primary (K-5) and b) secondary (Grades 6-12) ?

In summary …

- Teacher's role is vital to model ways of talking and dialogue
- Children can learn to reason more effectively in discussion with guidance and practice
- Productive dialogue in the classroom has a positive impact on learning outcomes
- Use a variety of strategies to teach key terms and vocabulary

So what does this have to do with computing and programming education? And to PRIMM?

So …. focus of my talk today …

1. How should we talk to and with learners in programming lessons to improve learning outcomes?
2. How does the PRIMM approach support ways of talking in the classroom?

Overview of seminar

Setting the scene: language and dialogue in the classroom

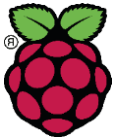
Learning programming: the challenges

The principles underlying the PRIMM approach

PRIMM and learning outcomes

PRIMM and vocabulary and talk

What's next?



Where can we go wrong teaching programming?

“Copy this code ..”

Takes a long time, introduces errors in syntax, and students often don’t understand what the code is supposed to do. Can take the whole lesson!

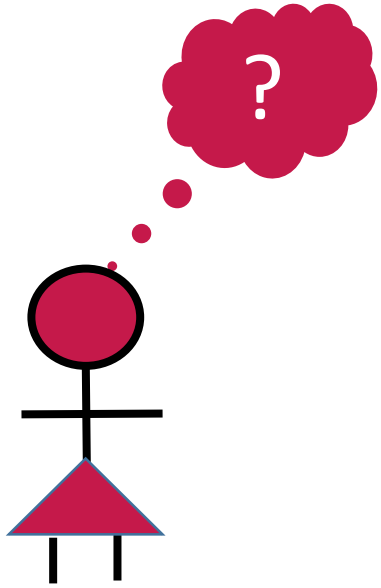
“Solve this problem by writing a program…”

Students are presented with a blank editor window, don’t know where to start, get frustrated by initial errors …. and give up ….

From model to make

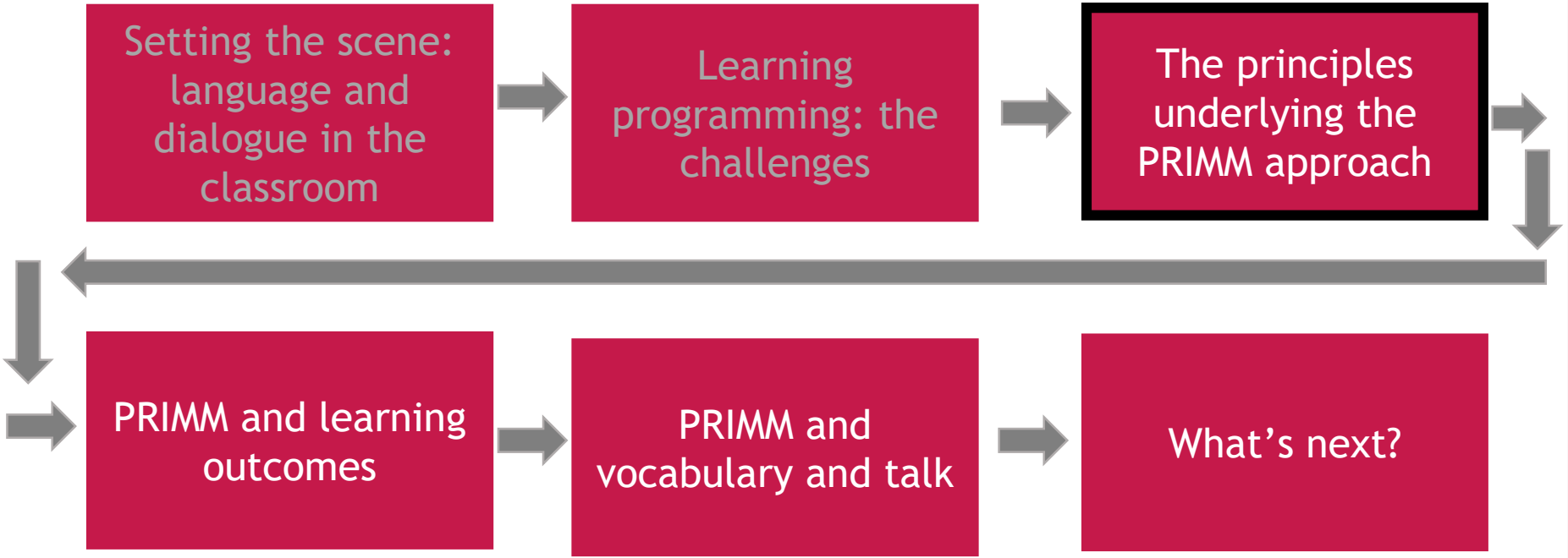
Watch me and then you do – “huge chasm”

What can cause learners to struggle with programming tasks?



Challenges faced	Possible reasons …
Perception of programming as hard – challenges resilience and self-efficacy	• Programs usually don't work until many iterations have passed • There is a "hard" myth (or is it a myth) • Belief in geek gene • Use of "adult" teaching approaches
A working program does not equate to understanding of programming concepts	• Too much copying of code without understanding • High cognitive load involved in programming • Focus on product rather than process
Students have misconceptions & incorrect mental model	• Many misconceptions are known but this has not transferred into teacher training • Teachers have fragile knowledge & misconceptions too

Overview of seminar



Key principles of PRIMM

- Read code before you write code
- Work collaboratively to talk about programs
- Reduce cognitive load by unpacking and understanding what program code is doing
- Use existing starter programs that the learner is not responsible for (especially if they don't work! – reduces emotional load)
- Gradually take ownership of programs when ready
- Structure lessons in a way that supports learning

Vygotsky (1896-1934)

- social interaction plays a critical role in children's learning.
- mediated activity promotes higher mental processes
- three major forms of mediation: material tools, psychological tools (including language), and interaction with other human beings.

The PRIMM approach

- **Predict** – given a working program, talk about it in pairs or groups. What do you think it will do?
- **Run** – run it and test your prediction
- **Investigate** – get into the nitty gritty. What does each line of code mean? Lots of activities to try here: trace, annotate, explain, talk about, identify parts, etc….
- **Modify** – edit the program to make it do different things
- **Make** – design a new program that uses the same nitty gritty but that solves a new problem

Other programming pedagogy & PRIMM

Some examples:

- Pair programming (Williams & Kessler, 2002)
- Worked examples (Morrison, Margulieux & Guzdial, 2015)
- Peer instruction (Porter et al, 2011)
- Live coding/ modelling (Rubin, 2013)
- Tracing/ reading code (Lister, many)
- Use-modify-create (Lee et al, 2011)
- Stepwise explanations (Wirth, 2001)

In this context, PRIMM is **one** approach to use for programming pedagogy

Specifically it is designed to:

1) Help teachers structure lessons
2) Reflect on the teaching process
3) Cater for varying needs

A Predict activity – you will need a piece of paper and a companion

Look at this Python (turtle) code

[Discuss with the person next to you]

Draw the output

```
from turtle import *

def square():
    for counter in range(4):
        forward(100)
        right(90)

square()
left(45)
square()
```

What discussions did you have?

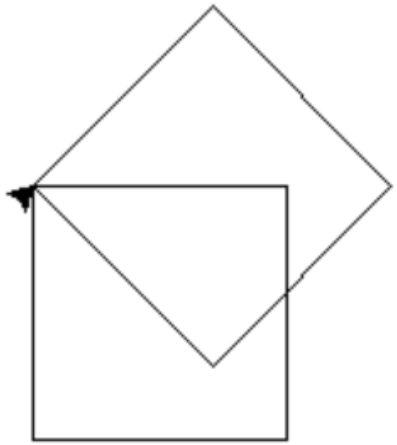
Did you help each other?

What did you learn?

```
from turtle import *

def square():
    for counter in range(4):
        forward(100)
        right(90)

square()
left(45)
square()
```



Now I will give you the program to run

This is what would actually be output

Did you get it right?

If not, what is different about your answer?

PRIMM Examples

Predict

Run

Investigate

Modify

Make

```
def cooking():
    print("Meal planner")
    print()
    print("1. Chicken curry ")
    print("2. Veggie lasagne")
    print("3. Burger and salad")
    print()
    print("Which of these meals is your favourite? (1, 2 or 3) ")
    answer = input()
    if answer == "1":
        print("Chicken curry coming up")
    elif answer == "2":
        print("Veggie lasagne coming up")
    else:
        print("Burger and salad coming up!")
    print("Enjoy!")


cooking()
```

Lesson 3

PRIMM Examples

Predict

Run

Investigate

Modify

Make

Task 1: Predict In pairs, look at the program below and write out what you think might happen when it runs.

```
def starter():
    number = 0
    while number <= 5:
        print("Hello")
        number = number + 1
    print("Goodbye")

starter()
```

What would you expect the computer to do? Write the output exactly as you think it will appear.

Task 2: **Run** Download and run the program and see if it does what you think it might do. You will find it at <insert your shared drive here>

Did the program run as you predicted? _____

What were the differences? _____

Keep starter programs on a shared drive

Students should download, check what they do and compare with prediction

PRIMM Examples

Predict

Run

Investigate

Modify

Make

Ask different types of questions

Task 3: In pairs, work out the answers to the following questions by examining the code and running it a few times.

1. What happens if you don't add any toppings?

.....

2. If you run this program and the user wants to add pineapple, olives and mushrooms, fill in the following table.

next_topping	toppings
	and pineapple
olives	
mushrooms	

3. Why do you need a while loop?

.....

4. What would happen if you didn't have the line next_topping = ""? (try it and see)

.....

.....

PRIMM Examples

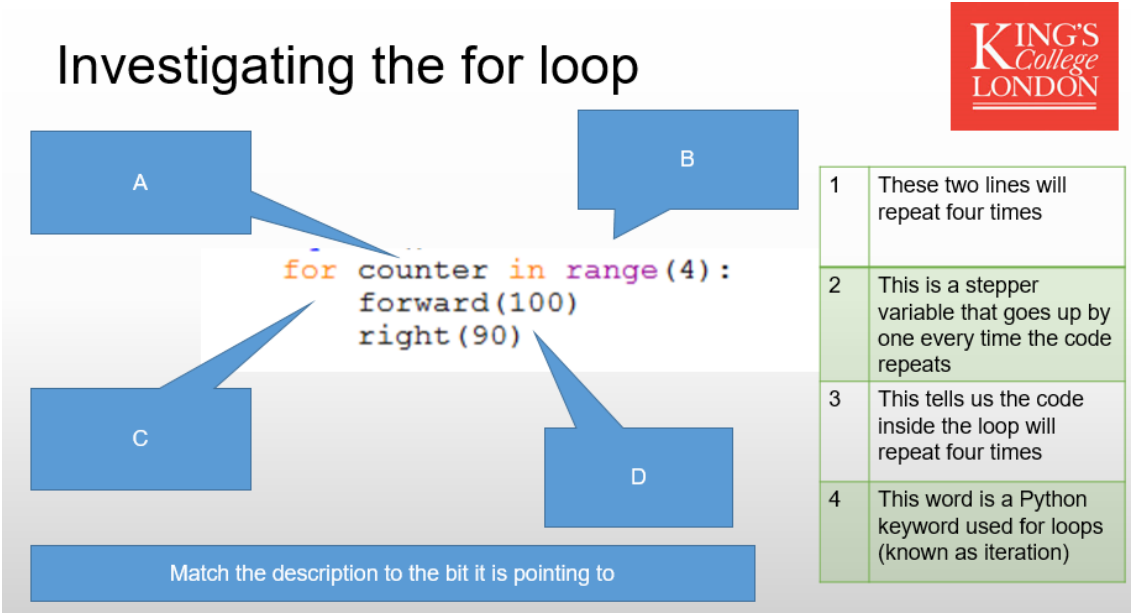
Predict

Run

Investigate

Modify

Make



THE BLOCK MODEL

(taken from Hello World Issue 14: The I in PRIMM)

(M) Macro structure			2. Ask: "What would happen if the input to the program was ___?"
(R) Relationships	5. Ask students to identify the scope of a variable	4. Draw the flow of control on the program	
(B) Blocks	3. Ask students to draw on the program to identify blocks of code or types of construct	1. Ask: "What would happen if those two lines were the other way around?" 4. Draw the flow of control	
(A) Atoms			6. Identify the purpose of a single statement
	(T) Text surface	**(P) Program execution**	**(F) Function**
	Architecture/Structure		**Relevance/Intention**

The Investigate phase can be mapped against the Block Model to ensure that all aspects of program comprehension are covered

PRIMM Examples

Predict

Run

Investigate

Modify

Make

Improve the pizza program so that it doesn't print out "and X" at the end. You will have to add an "if" statement to do this.

Modify the program to have a conversation with somebody about sport. An example is given below but you can add your own questions.

Change your triangle function so that it uses a for loop

Write a function pentagon() to create a pentagon (5 sides) with sides length 200

Write a function hexagon() to create a blue hexagon (6 sides) with sides length 50

PRIMM Examples

Predict

Run

Investigate

Modify

Make

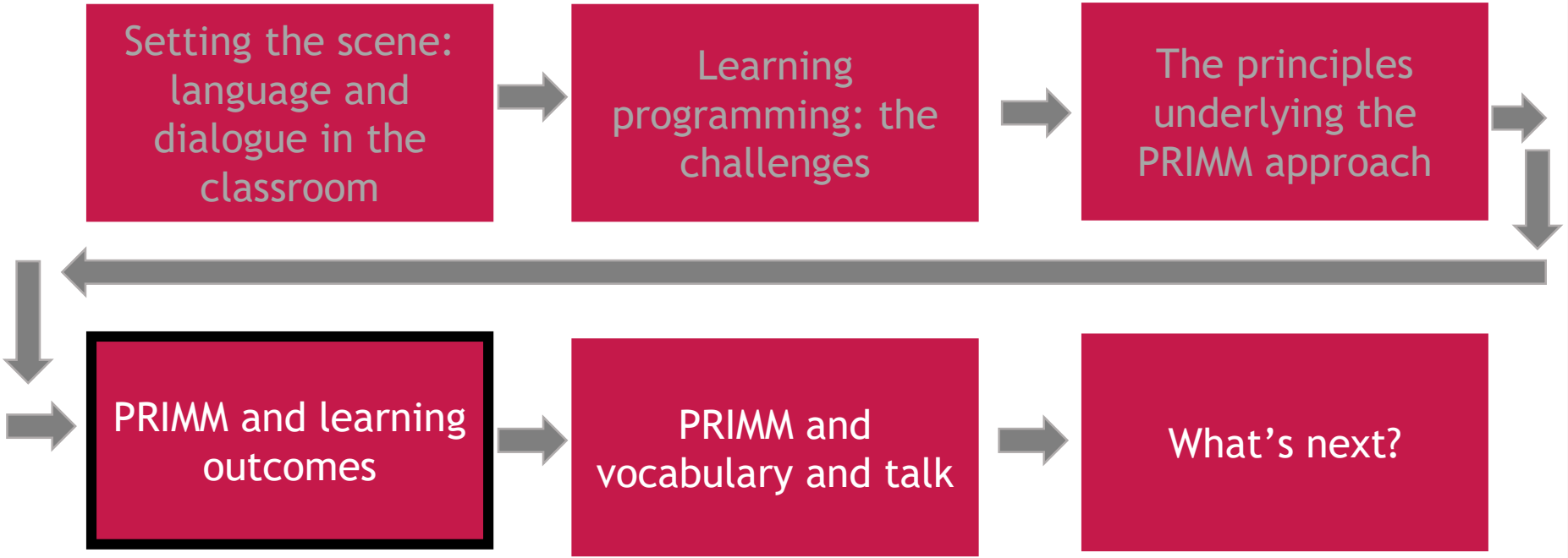
Using your Python skills so far you are going to develop a quiz of 5 questions about Geography

1. First write down 5 questions and 3 possible answers to each
2. Then write your program
3. Test all the different options

You will be scored on whether it works, whether you have good questions, and how well it is presented (spacing, grammar etc.)

2. Write a version of the game "I went shopping and .. " . Then try it on somebody who can't see the computer to see if they can remember everything the computer has stored from the shopping game? The game should stop when the user presses the enter key rather than adding another item.

Overview of seminar



Our research

Pilot study 2017

6 teachers
80 students (aged 11-15)
4-7 lessons
Activity sheets
Teachers edited materials

"It was amazing! In one lesson, they pretty much all got the concept of a function"

Main study 2018

Mixed methods approach
13 teachers
493 students PRIMM
180 in control group
Baseline & Post test
10-12 weeks
Interviews of teachers
Focus group & journals

Main study: students

Two groups of students:

- 493 were taught using 10 PRIMM lessons
- 180 students formed a control group

Both groups completed baseline test and post test (not equivalent) of programming questions

Baseline test was used to test for equivalency of both groups

Groups were compared by their scores in the post intervention test

Main study: teachers

- Teachers were selected who were teaching 11-14 year olds selection and iteration in Python for a term (~10 – 12 weeks)
- 13 teachers (out of 14 signed up) attended a day's training on the PRIMM approach and how to use the materials
- Teachers kept journals which they completed after each lesson (some did)
- 9 teachers were interviewed by one researcher at the end of the period
- An online focus group was held at the end of the study for teachers to discuss experiences together

Results - students

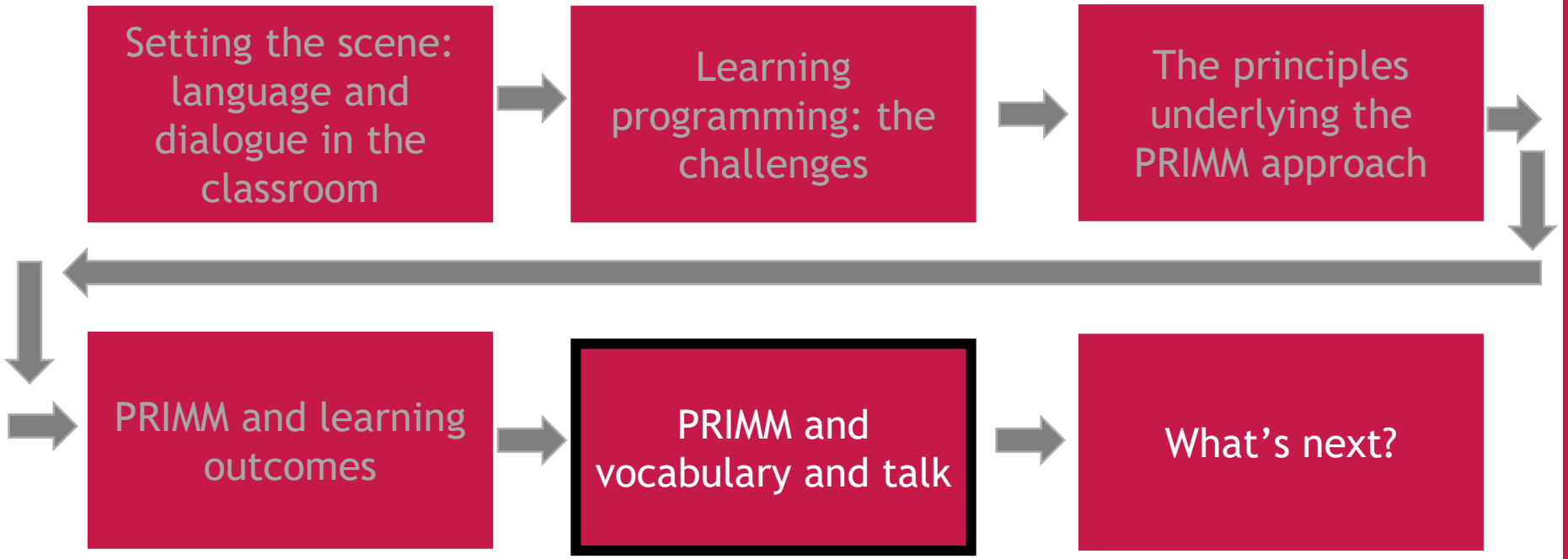
Results

- Control group and trial group were equivalent in the pre-test
- Experimental group performed better than control group in the post test

Table 4. Descriptive statistics.

Control	Baseline test	Posttest	Experimental	Baseline test	Posttest
Mean	4.58	2.575	Mean	4.89	3.284
Median	4.0	2.0	Median	5.0	3.0
Std. Deviation	1.911	2.1916	Std. Deviation	2.115	2.510
Skewness	.548 (se = .181)	1.237 (se = .181)	Skewness	0.88(se = .110)	.873(se = .110)
Gray Kurtosis	.301 (se = .360)	1.314 (se = .336)	Kurtosis	−.389(se = .220)	.301(se = .220)

Overview of seminar



Planning a lesson using PRIMM

Predict-Run-Investigate-Modify-Make

Lesson structure

- PRIMM fosters structure
- Routine becomes familiar
- Educators adapt to students' needs
- Each step can be further differentiated

Language/ talk

- Students practise using appropriate programming terms
- Misconceptions can be articulated and explored
- Collaboration is a key element of PRIMM

Content/ questions

- When carefully selected questions, help students explore the program
- Should be within student's ZPD (zone of proximal development)
- Teacher is key in making content relevant.

Shared artefacts

- Programs first presented on the social plane ("not mine")
- Giving students a program to run (not copy) reduces anxiety
- Gradually student takes ownership ("mine")

image from http://teachcomputing.org/pedagogy - recent Quick Read on PRIMM

PRIMM and Classroom Talk

- Talking about a program and how it works helps learners to find the right vocabulary to use to articulate their understanding. Having a common language to talk about programming constructs is important.
- Actually verbalising out loud the steps of a program that is difficult to understand can help learners to focus on atomic or smaller elements at a time.
- Through dialogue with others, we can ask and answer questions, and learn from others.
- With PRIMM, the teacher facilitates these processes – encouraging discussion, modelling vocabulary use, asking in-depth questions

*“ It was the fact that they were talking and **bouncing ideas off each other** made it enjoyable and different.” (teacher, main study)*

*"There was certainly more **active talking** and **planned talking** about the programming because of the way that the questions are worded in the worksheets and the resources"*

(Teacher, main study)

"I noticed a big difference in terms of the girls collaborating with each other and trying to sort out each other's problems.

Usually that's been confined to one or two girls who feel quite confident, but with PRIMM with the tasks that they were doing, they always felt that they were closer to a solution that they might necessarily have felt in the past when I've taught them.

So they were actively engaging in helping each other and looking at each other's code and ***making suggestions***"

(Teacher, Main Study)

Our research

Pilot study 2017

6 teachers
80 students (aged 11-15)
4-7 lessons
Activity sheets
Teachers edited materials

Main study 2018

Mixed methods approach
13 teachers
493 students PRIMM
180 in control group
Baseline & Post test
10-12 weeks
Interviews of teachers
Focus group & journals

New Study 2020/2021

Use of language in PRIMM

- Current study (2020-2021)
- Research questions
  1. In what way does talk support the improvement of programming skills?
  2. How can teachers facilitate effective classroom dialogue about programming?
- Stage 1 (complete) : Interviews with 19 experienced teachers who have used PRIMM for some time in the classroom. Very early themes identified
- Stage 2 (2021) : Reflective talk diaries, using PRIMM as the teaching approach because it naturally involves lots of talk

Theme 1: Students finding a language to express themselves…

"They really get stumped with the difference between a parameter and an argument. And understanding what a parameter is and a return value. I think that's where you'd see the most difficult thing that they talk about. Which is why we try and give them a language because the language helps them to express themselves better when they're talking about it. And also it helps them I think to have a mental model of what that is, if you give it a name." (secondary teacher)

Theme 2: Students engaging in productive discussion…

The children are generally talking to each other and often working through some questions together, deciding what they want to say as their answer. Then there's a lot of discussion around how to get things to work [primary teacher]

Particularly at the beginning of the PRIMM lesson or series of lessons, there's very much a sense of discovering together what it does and how it works and so an awful lot more peer conversation going on. [primary teacher]

Theme 3: Teachers modelling programming terms…

I'm very strict on vocabulary on class so I tend to model quite a lot from the front and model the correct vocabulary … I think PRIMM helps that when you're doing "what do we think this does" and then you can go and talk about it, insisting that they talk in the correct vocabulary. And then just letting them have those conversations with each other. (secondary teacher)

Theme 4: Talking less … and at a more advanced level …

" I find that I'm having less time talking to the whole group, because we can get the programs running in the first place, so I'm talking to the group probably less … but … I'm talking at a more advanced level to the whole group, but for less time. When I'm asking questions, they're usually much more useful and probing questions … " [secondary teacher]

RECAP! Remember this …

Dialogue leading to learning outcomes

In maths, literacy and science classrooms, Howe et al (2019) found three aspects of teacher-student dialogue strongly predicted performance in assessments:

1) **Elaboration** - where building on/elaborating /evaluating/clarifying of a previous contribution was invited or provided
2) **Querying** - where a previous contribution was doubted/challenged/rejected
3) **Student Participation -** where across the lesson multiple students were seen to engage with each other's ideas, and not merely respond to their teacher's questions.

Dialogue leading to learning outcomes – teachers' comments

Elaboration

"If you put it out there, there'll be somebody in the classroom who says, well, I know what this bit does, and someone will say, well, I know what this bit does, and so it's bringing those things together. Even very young children can do that, so it's quite interesting." (primary teacher)

Querying

"But also, you can pick up the fact that the questions they ask each other do change. Well, I want to do that, how? Where does that come from? They'll have that conversation with each other." (secondary teacher)

Participation

"Or even peers of the same ability who are having a problem, they can actually discuss that together much better because they… What's that thing there? They can actually have that, well, I've tried this, have a look at…" (secondary teacher)

Initial findings

- Teachers report that learners taught using PRIMM are developing a language to talk coherently about their programs
- Teachers use the Predict and Investigate stages to ensure aspects of the program are labelled verbally
- Teachers use different approaches to introduce vocabulary and model its use.
- Teachers generally use the PRIMM structure to facilitate discussion about how a program works
- Teachers generally encourage a common language to talk about programming.

Returning to original questions:

1. How should we talk to and with learners in programming lessons to improve learning outcomes?

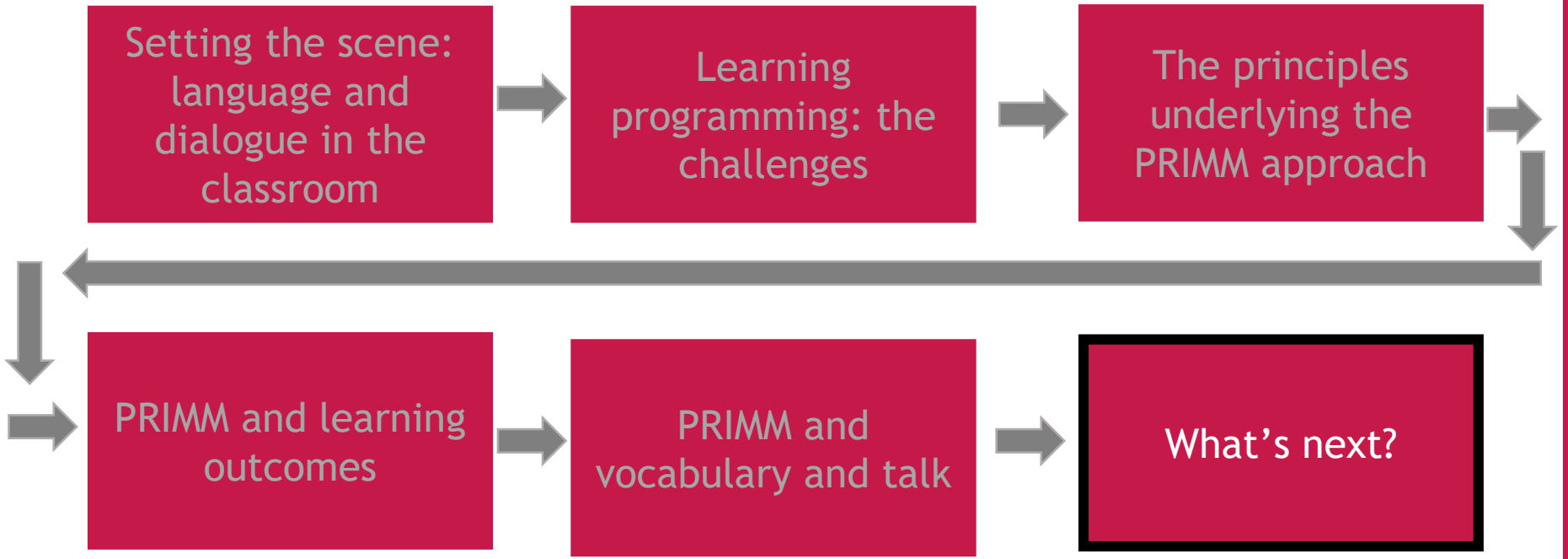
- Encourage students to explain how a program works (or doesn’t)
- Model use of programming construct vocabulary to create a shared language
- Use Predict and Investigate stages to talk about the program, using dialogue and key vocabulary

Returning to original questions:

2. How does the PRIMM approach support ways of talking in the classroom?

- The focus on investigating and understanding the code means that learners need to articulate what it does
- Emphasis on pair/group work means that students have an opportunity for dialogue
- Focusing the dialogue on the program itself means that key vocabulary items are introduced and frequently reinforced

Overview of seminar



What's next for PRIMM?

Language and Talk

- Further work to investigate how PRIMM, and similar approaches, encourage productive talk in the classroom
- Taxonomy of different types of talk to support teachers using PRIMM

General PRIMM

- Continue work on questioning (around the Block Model)
- Something between the 2 Ms!
- Is PRIMM a semantic wave?

In summary

I've talked about …

- Language in education
- Why programming is difficult (and we don't always teach it well)
- Key principles underlying the PRIMM approach
- PRIMM and learning outcomes
- PRIMM and language and talk

Further reading

Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, *29*(2-3), 136-176.

Further browsing

http://primming.wordpress.com

Thank you for listening!

Contact me at

sue@raspberrypi.org
or @suesentance

Questions for breakout groups

Teachers?

How do you encourage talk around programming?

[… and in a pandemic?]

Researchers?

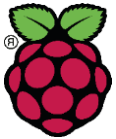
Why are we not researching language in programming lessons more?

New to PRIMM?

Why does PRIMM support learning in programming lessons?

New to K12/ school?

How is teaching children programming different to teaching adults?



Some references

- Ruiguo Cui & Peter Teo (2020): Dialogic education for classroom teaching: a critical review, Language and Education
- Diethelm, Juliana Goschler and Timo Lampe , Language and Computing, in Sentance, S., Schulte, C. and Barendsen, E. Computer Science Education: Perspectives on learning and Teaching in School. Bloomsbury Academic.
- Neil Mercer & Claire Sams (2006) Teaching Children How to Use Language to Solve Maths Problems, Language and Education, 20:6, 507-528
- Martin Nystrand , Lawrence L. Wu , Adam Gamoran , Susie Zeiser& Daniel A. Long (2003) Questions in Time: Investigating the Structure and Dynamicsof Unfolding Classroom Discourse, Discourse Processes, 35:2, 135-19
- Paul J. Riccomini, Gregory W. Smith, Elizabeth M. Hughes & Karen M. Fries (2015) The Language of Mathematics: The Importance of Teaching and Learning Mathematical Vocabulary, Reading & Writing Quarterly, 31:3, 235-252
- Sentance, Sue, Jane Waite, and Maria Kallia. "Teaching computer programming with PRIMM: a sociocultural perspective." *Computer Science Education* 29.2-3 (2019): 136-176.