Nested Loop with Define Procedures

**Computing Science Glossary**

**Procedure** Grouped code that can be used lots of times by referring to the name

**Completion Problem** Some code is provided but not all code is connected so pupils have to complete the program

**PRIMM** A strategy that promotes Predicting, Running, Investigating and Modifying code before Making something.

**IPRIMM** Stages of PRIMM but in a different order

**Algorithm** Part of planning stage before programming written for another human to read.

**Code** Written for a digital device

**USE MODIFY MAKE** A strategy that promotes using and modifying code before creation

**Loop** Instructions that are repeated more than once

**Nested loop** One or more loops inside each other

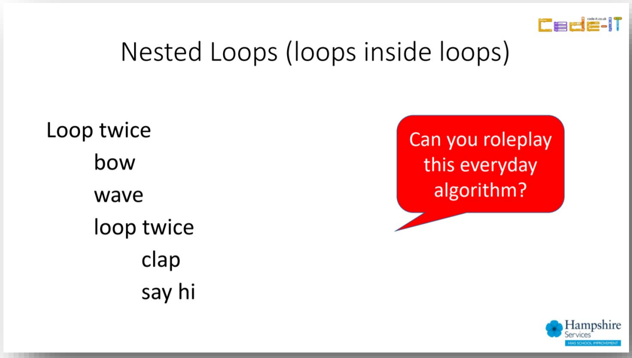
# Exploring Nested Loops with Procedures

## Questions & Answers

**How does this fit in with other shape programming?** See overview document **What age is this for?** KS2 but the best place to start if not programmed using nested loops before. **How hard is this to teach?** Very easy as all the code and question are in the booklets. **How do we Assess learning?** Pupils use answers provided to mark their own work. **Is it in line with NC?** Yes see next page. **Why is there a choice?** There are lots of ways of teaching programming and no one really knows which are best or better or even if there is a best or better for all. We do know that it helps pupils to encounter a variety of different types of method so they are continually challenged. If you work your way through all modules I recommend you switch strategies each time to keep the challenge high. **How creative is this?** This combines the best knowledge we have about how to learn something new with the opportunity at the end for pupils to create something that they want to create that uses their new knowledge.

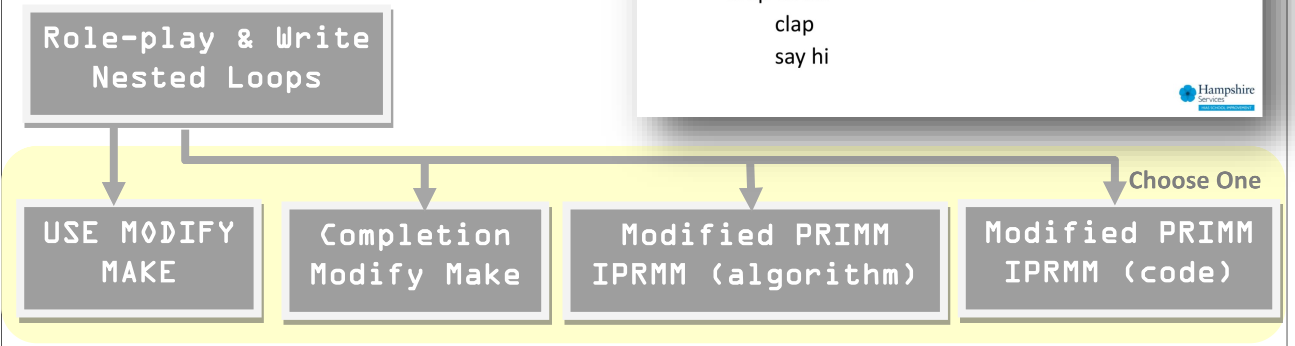
## Role-play & write nested loops

**Download Concepts before coding PDF** from http://code-it.co.uk/wp-content/uploads/2019/04/conceptbeforecoding.pdf Follow the links in the menu to nested loops. Use slides (40-48) to roleplay and write simple fun procedures.


Nested Loops (loops inside loops)

```
Loop twice
    bow
    wave
    loop twice
        clap
        say hi
```

Can you roleplay this everyday algorithm?


Role-play & Write Nested Loops
Choose One
USE MODIFY MAKE
Completion Modify Make
Modified PRIMM IPRMM (algorithm)
Modified PRIMM IPRMM (code)

## Booklet Choices

You choose from one of the options above. Each option has its own booklet which guides pupils through the stages, making them think deeply about either the code or the algorithm before modifying it and having a choice of things to make. Pupils are instructed when to work in pairs and when to work alone. Cut up the answer sheet (last page of the booklet) into sections as pupils will need to mark their work as part of their learning process.

## National Curriculum Programs of Study

(bold text is covered in this module)

**Pupils should be taught to:**

**design, write and debug programs that accomplish specific goals**, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

**use sequence,** selection, and **repetition in programs;** work with variables **and various forms of input and output**

**use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs**

## Before the module

Read the planning and download the PDFs for role-play and write nested loops. Decide which booklet variation you are going to use and download and print it out one per pupil. Remove the answer sheets for pupils and strim them into sections for pupils to access when they need to. Download the code for your version of Scratch 2 or 3 and place it on your network where pupils can access it, or note where it is on the Scratch website if using Scratch online

## Formative assessment support

Asking pupils to read code or algorithm out loud helps pupils who are struggling to predict, investigate or modify.

If pupils are struggling to work together in a meaningful way then encouraging and rewarding positive attitudes to working collaboratively using the communicates stickers shown at the end helps.

Lots of misconceptions can be solved by reading the code or algorithm slowly and out loud to their partner.

## Classroom Organisation

In some sections pupils are asked to work with a partner of similar programming ability. If you are not sure what programming ability they are go with Maths skills as a starting place. Move partners around between modules so that pupils benefit from different interactions.

## Assessment

You can get pupils to mark their booklets giving one mark for each correct answer there is also a short Kahoot quiz that pupils can take. If your pupils complete this please consider anonymising the results, asking your head teacher for permission and emailing it to phil.bagge@code-it.me to help us determine which method works best.

## Resources

**Role-play & Write Nested Loops**

PDF Download Slides 40-48

(20 mins)

**Completion Modify Make**

Pupil booklets

Scratch 2 & 3 Code to download

Scratch 3 code on Scratch website

**PRIMM Algorithm & Code**

Pupil booklets

Scratch 2 & 3 Code to download

Scratch 3 code on Scratch website

**USE MODIFY MAKE**

Pupil booklets

Scratch 2 & 3 Code to download

Scratch 3 code on Scratch website

**All Resources at**

http://code-it.co.uk/goldshape/

## Further Research Reading

**Use Modify Create**

Irene Lee et al Computational thinking for Youth in practice (2011)

**PRIMM** Sentence

https://blogs.kcl.ac.uk/cser/2017/09/01/primm-a-structured-approach-to-teaching-programming/

**Four levels of abstraction**

This article includes an example of the four levels of abstraction and sign posts the work of Waite and Armoni

http://code-it.co.uk/algprogdiff/

I recognise there is more than one way to solve/describe a problem

I can evaluate my solutions against a set criteria

I can design criteria to evaluate my creations

I don't just accept the first solution

I look for a range of solution to the same problem

I look for how a project can be extended

Handles Ambiguity

Open Ended Problem Solver

I can break complex problems into parts

I can discover / concentrate on the most important part of a problem

Evaluates

Insert picture of your students here

Copes with Complexity

I can identify patterns in problems & solutions

I can contribute useful ideas to a partner or group

I can encourage others to share their ideas

Communicates

Adapts

I can adapt existing ideas to solve new problems

I can develop, test and debug until a product is refined

I lead using all the people talent in my group

I learn from setbacks and don't let them put me off

I can persevere even if the solution is not obvious

Perseveres

Investigates

I make predictions about what will happen

I repeatedly experiment through predicting, making, testing & debugging

@baggiepr

Inspired by Behaviour Rubric created with @MarkDorling and linked at http://code-it.co.uk/attitudes/