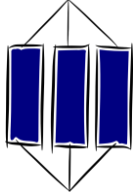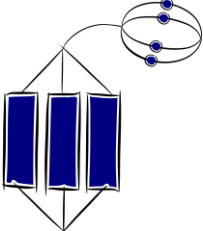**HookED**
www.pamhook.com

Using Scratch MIT to **[insert programming task]** Possible components: • add a new sprite • make a sprite move • make a sprite speak • change the background • edit sprite costume • add audio file • add a green flag control • make two sprites interact (bounce, wait, if) • make a sprite respond to controls (mouse or keyboard) • add sound effect to a specific sprite action • add background music that starts when the green flag is clicked • rolling background • apply variables for acceleration and smooth movement etc. • problem solve - resolve bugs in scripts. (Components from Twitter @daibarnes)	**Prestructural**	**Unistructural**	**Multistructural**	**Relational**	**Extended Abstract**
	Lacks knowledge of the assessed component – may have pieces of unconnected knowledge but make no sense to the student	Focuses on a single aspect – has an understanding of one aspect but not of its significance to the whole or relationship to other aspects	Focuses on several separate aspects – has an understanding of more than one aspect but not of their significance to the whole or relationship to other aspects.	Relates different aspects together.	Seeing the concept from an overall viewpoint.
	Learning outcomes show unconnected information, no organisation. *E.g. "I need help or direction"*	Learning outcomes show simple connections but importance not noted. *E.g. "I will have a tilt at it"*	Learning outcomes show connections are made, but significance to overall meaning is missing. *E.g. I will use trial and error to find a solution"*	Learning outcomes show full connections made, and synthesis of parts to the overall meaning E.g. "I plan to do X because it will … I know what to do and why .."*	Learning outcomes go beyond subject and makes links to other concepts - generalises *E.g. "I sense what to do to find the best solution ….I seek feedback and adjust my actions in response "*
	I need help to make sense of the programming task.	I can write code to [insert task] if directed.	I can create code to [insert task/s], but I make mistakes. I can create code to [insert a range of tasks] to solve a problem. But I am not always sure why, when or where I should use the code. I need help to identify my mistakes.	I can create correct code to solve a problem by selecting and carrying out a logical sequence of tasks/steps. I know when and where I need to use [insert a range tasks] to solve a problem. I can relate my findings using appropriate programming statements. I can recognise where I have made a mistake and correct it.	I can make connections beyond the scope of question and can transfer knowledge to a new situation I can seek feedback to improve the efficiency of my code.

Example	Insert Scratch code here				
Effective strategies	Teacher and student strategies				