**Highsted Knowledge Organiser**

**Computer Science: Scratch Programming – Year 8 – Term 1**

## What I need to know

- What is scratch programming
- What are the different blocks in scratch
- How to write a simple program to interact with the sprite
- How to program a virtual pet game
- To test and debug scratch code

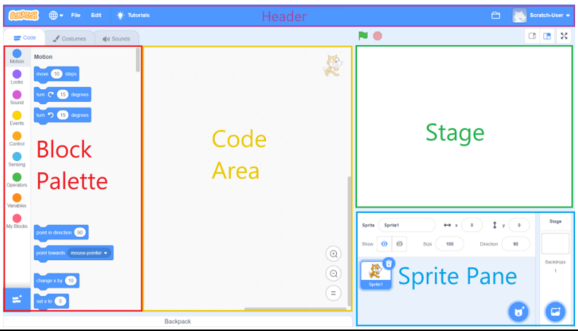
## Key Vocabulary

| | | |
|---|---|---|
| - Blocks | - Stage | - Sprites |
| - Costume | - Condition | - Iteration |
| - Variable | - Green flag | - Loop |
| - Algorithm | - Editor | - Script |

## Student reference point

### What is a scratch programming?

Scratch programming is a language where you can create interactive games and animation and for learning coding in general. The scratch interface is divided in 4 main sections.




### The 3 key programming concepts

1. **Sequence**

A sequence is a set of code that is run/executed in order

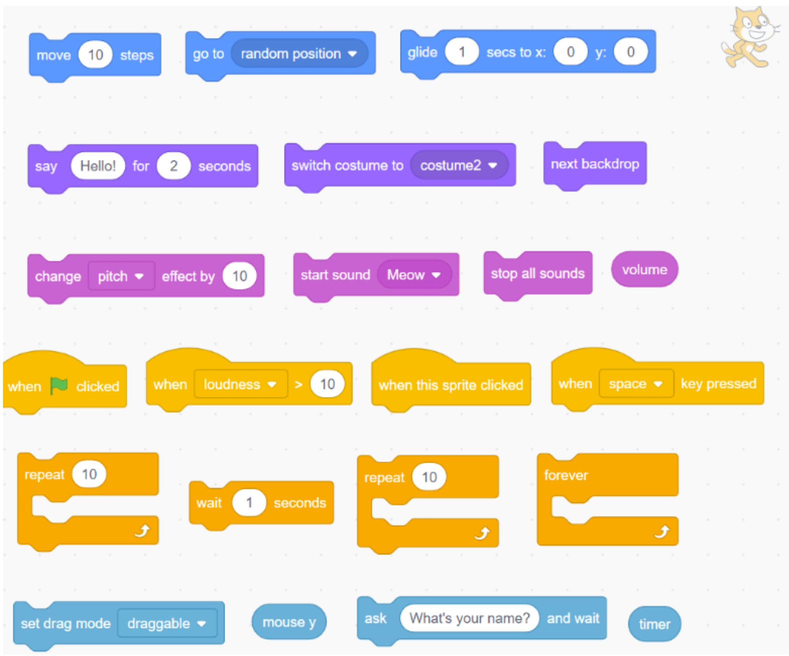
2. **Selection**

A set of codes that employs decisions between multiple choices (makes decision)

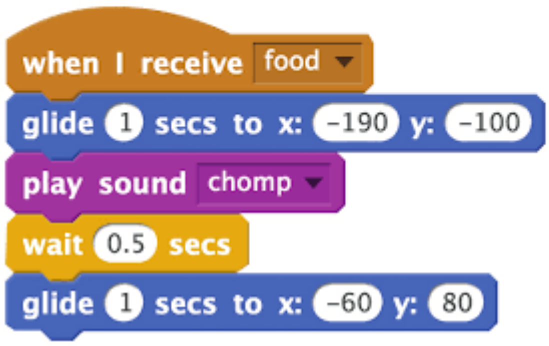
3. **Iteration/Loop**

A set of codes that is repeated over a number of times.

### Scratch Blocks




### Virtual Pet Code Example




## Challenge question

- What are variables and why do we use them in programming?

## Suggested reading

- https://scratch.mit.edu/starter-projects

# Highsted Knowledge Organiser

## Computer Science: Computational Thinking and Algorithms – Year 8

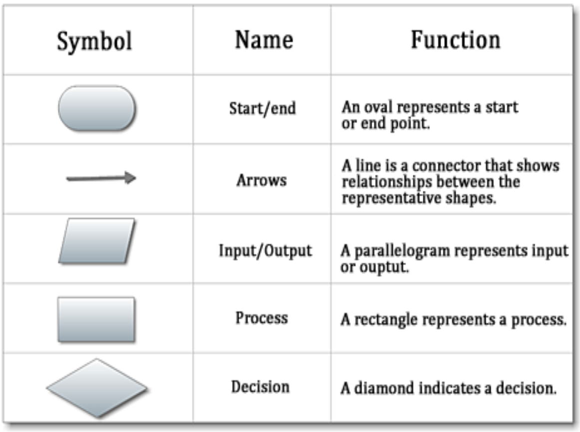
### What I need to know

- What are the 4 techniques of computational thinking
- What does the term algorithm means
- How to represent algorithms (flowchart and pseudocode)
- Create algorithms using flowchart diagram

### Key Vocabulary

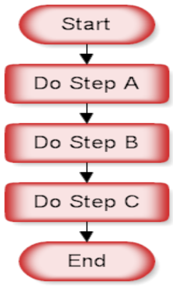
| | |
|---|---|
| - Algorithms | - Abstraction |
| - Flowchart | - Pattern recognition |
| - Pseudocode | - Decomposition |
| - Sequence | - Selection |
| - Loops/Iteration | - |

### Student reference point



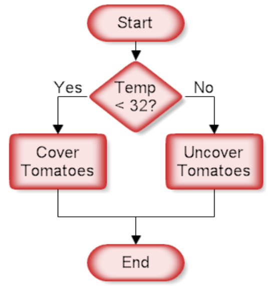
| Symbol | Name | Function |
|---|---|---|
| | Start/end | An oval represents a start or end point. |
| | Arrows | A line is a connector that shows relationships between the representative shapes. |
| | Input/Output | A parallelogram represents input or ouptut. |
| | Process | A rectangle represents a process. |
| | Decision | A diamond indicates a decision. |

**Sequence**



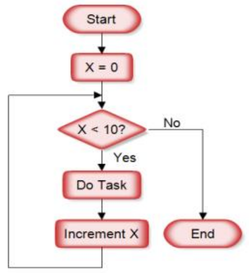

**Start**
**Do step A**
**Do step B**
**Do step C**
**End**

**Selection**

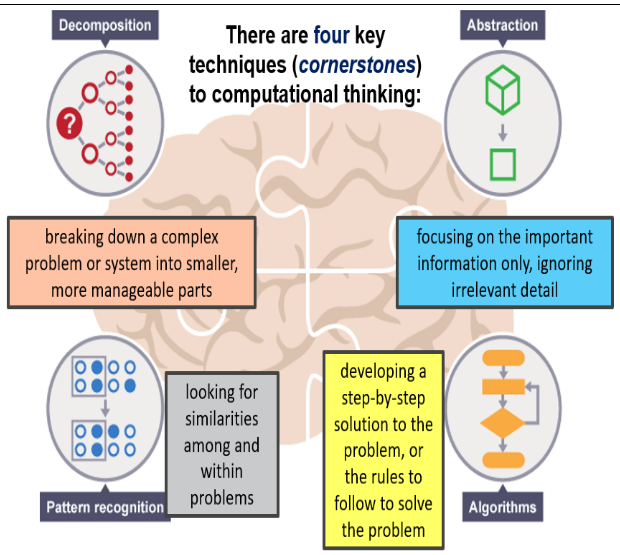



**Start**
**If Temp < 32 Then**
**Cover Tomatoes**
**Else**
**Uncover Tomatoes**
**End**

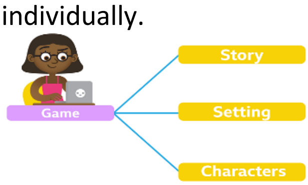
**Iteration**




**Start**
**X = 0**
**While X < 10**
**Do task**
**X = X + 1**
**End**

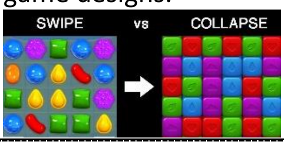



**Decomposition** in game design helps the team to look at different parts of the game individually.




**Abstraction** in game design helps the team to look and program for what is necessary to the game only.

**Pattern Recognition** in game design help the team to not recode or redesign everything from scratch but use ideas or template from other similar game designs.




**Algorithm** in game design helps the team to write accurate instruction on how to play the game.

Calculating the average of scores

1 Pseudcode Algorithm:

**Display** " Enter the bowler's score for the 1st game:".
*Read* score1.
**Display**" Enter the bowler's score for the 2nd game:".
*Read* score2.
**Display**" Enter the bowler's score for the 3rd game:".
*Read* score3.
averageScore = ( score1 + score2 + score3 )/3
**Display**"The bowler's average score is:".
**Display**averageScore.

### Challenge question

- How can Computational Thinking be used to help programmers build a computer system?
- Create a flowchart to find the sum of 5 numbers.

### Suggested reading

- https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1
- https://www.bbc.co.uk/bitesize/guides/zpp49j6/revision/3
- https://www.bbc.co.uk/bitesize/guides/z7kkw6f/revision/6