Programming Concepts and Skills Supported in Scratch

In the process of creating interactive stories, games, and animations with Scratch, young people can learn important computational skills and concepts.

Problem-Solving and Project-Design Skills

- logical reasoning
- debugging problems
- developing ideas from initial conception to completed project
- sustained focus and perseverance

Fundamental Ideas about Computers and Programming

- computer programs tell the computer precisely what to do, step-by-step
- writing computer programs doesn't require special expertise, just clear and careful thinking

Specific Programming Concepts

Concept	Explanation	Example
sequence	To create a program in Scratch, you need to think systematically about the order of steps.	when space key pressed go to x: -100 y: -100 glide (2) secs to x: (0) y: (0) say Let the show begin! for (2) secs play sound fanfare until done
iteration (looping)	forever and repeat can be used for iteration (repeating a series of instructions)	play drum 54 T for 0.2 beats move 10 steps turn 10 degrees
conditional statements	if and if-else check for a condition.	if x position > 200 set x to -200 wait (0.1) secs
variables	The variable blocks allow you to create variables and use them in a program. The variables can store numbers or strings. Scratch supports both global and object-specific variables.	when clicked set score to 0 forever move 10 steps if touching color 2 change score by 1
lists (arrays)	The list blocks allow for storing and accessing a list of numbers and strings. This kind of data structure can be considered a "dynamic array."	add bread to food will add red apples to food will set counter with 1 repeat length of food will for 2 secs change counter with 1

event handling	when key pressed and when sprite clicked are examples of event handling – responding to events triggered by the user or another part of the program.	when left arrow key pressed point in direction -90 move 10 steps
threads (parallel execution)	Launching two stacks at the same time creates two independent threads that execute in parallel.	when clicked glide 3 secs to x: -75 y: 80 glide 5 secs to x: 175 y: -130 when clicked forever next costume wait 1 secs
coordination and synchronization	broadcast and when I receive can coordinate the actions of multiple sprites. Using broadcast and wait allows synchronization.	For example, Sprite1 sends the message winner when condition is met: wait until score > 100 broadcast winner This script in Sprite2 is triggered when the message is received: when I receive winner This sound cheer This sound cheer This sound cheer This say You won the game!
keyboard input	ask and wait prompts users to type. answer stores the keyboard input.	ask What's your name? and wait say join Hello, (answer)
random numbers	pick random selects random integers within a given range.	set x to pick random -100 to (100)
boolean logic	and, or, not are examples of boolean logic.	when space v key pressed if touching color ? and x position > 200 change score v by 1 play sound music v until done
dynamic interaction	mouse_x, mouse_y, and loudness can be used as dynamic input for real-time interaction	forever set size to loudness * 4 % wait 0.01 secs
user interface design	You can design interactive user interfaces in Scratch – for example, using clickable sprites to create buttons.	when Sprite1 clicked change brightness effect by 25 play drum 48 for 0.2 beats change brightness effect by -25

Programming concepts <u>not</u> currently introduced in Scratch: procedures and functions; parameter passing and return values; recursion; defining classes of objects; inheritance; exception handling; file input/output.