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.

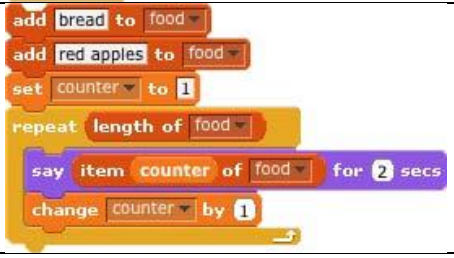
Problem-Solving and Project-Design Skills

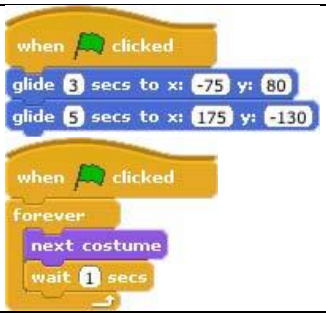
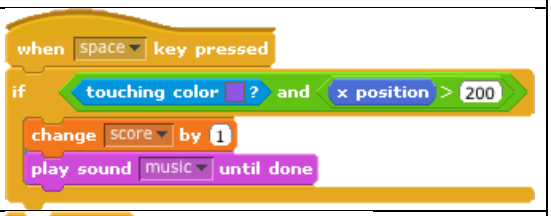
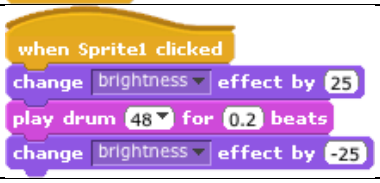
- logical reasoning
- debugging problems
- developing ideas from initial conception to completed project
- sustained focus and perseverance

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.	
iteration (looping)	**forever** and **repeat** can be used for iteration (repeating a series of instructions)	
conditional statements	**if** and **if-else** check for a condition.	
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.	
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."	

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.	
threads (parallel execution)	Launching two stacks at the same time creates two independent threads that execute in parallel.	
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:  This script in Sprite2 is triggered when the message is received: 
keyboard input	**ask and wait** prompts users to type. **answer** stores the keyboard input.	
random numbers	**pick random** selects random integers within a given range.	
boolean logic	**and, or, not** are examples of boolean logic.	
dynamic interaction	**mouse_x, mouse_y,** and **loudness** can be used as dynamic input for real-time interaction	
user interface design	You can design interactive user interfaces in Scratch – for example, using clickable sprites to create buttons.	

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.