Introduction to Computer Programming and Scratch 3.0

Scratch Basics:

- Sprite Programming - Use visual programming blocks to make sprites do things.
- Sequential Processing - Programmers need to think systemically, step-by-step, in order to program events in a logical order.
- Application Troubleshooting - Debugging programs to fix errors is on-going.
- Interface Design - Programmers can design interactive use interfaces, such as creating buttons using clickable sprites.
- Scratch applications are made up of ***Sprites*** (objects or characters) that interact with each other and are controlled and animated by scripts.
- ***Scripts*** are made up of 1 or more Scratch code blocks that are designed to perform a specific task or action.
- The Scratch program has more than 100 different Scratch blocks, each designed to fulfill a specific purpose. Blocks can be classified into 3 basic types:

  Stack Blocks – majority of Scratch blocks are Stack Blocks. Stack blocks are code blocks with a notch at the top or a bump at the bottom. The notches/bumps serve as visual indicators that identify how the blocks can be snapped together to create programming logic.

  Hat Blocks – Hat blocks have rounded or curved tops and a bump at the bottom, visually indicate that it can be snapped on top of other stack blocks. Hat blocks provide the ability to create event-driven scripts. An ***Event-driven script*** is one that automatically executes when a specified event occurs. The most commonly used Hat Block is the green flag block.

  Reporter Blocks – Reporter blocks are code blocks that have either rounded or angled side and are specifically designed as a mechanism for providing input for other code blocks to process.
- The 100 code blocks in Scratch are organized into ***8 different categories***:
  1. **Motion** = Colored blue; control sprite placement, direction, rotation & movement.
  2. **Looks** = Colored purple; affect sprite & background appearance & provide ability to display text, including Think & Say blocks.
  3. **Sound** = Colored pink; Control the playback & volume of musical notes & audio files.
  4. **Events** = Colored mint gold; sense events to trigger scripts to run.

5. **Control** = Colored gold; trigger script execution based on predefined events, repeatedly execute programming logic using loops, & perform conditional logic.
6. **Sensing** = Colored sky blue; used to determine the location of the mouse-pointer, its distance from other sprites, & whether a sprite is touching another sprite.
7. **Operators** = Colored green; perform logical comparison, rounding & arithmetic operations.
8. **Variables** = Colored orange; used to store data used by applications when they execute.
9. **My Blocks** = Colored red; allows the user to make their own stack blocks.

- Important Tips to Remember regarding code blocks

Code blocks that have a check-box to the left have the ability to display a monitor within the block on the Scratch stage.

i. A ***monitor*** is a small block that displays the value currently assigned to the code block, beginning with a default value. The way the monitor looks can be changed by right-clicking on it and selecting a different view/appearance format (e.g. large readout or slider bar).

To get **Help** regarding how to use a block, right-click on the block (was in version 2.0, but no longer available in version 3.0)

Beyond the Basics in Scratch:

Event Handling - When key pressed & When sprite clicked are examples of event handling (i.e. responding to events triggered by the user or another part of the program).

Program Synchronization & Parallelism - **Broadcast** & **When I Receive** can coordinate the actions of multiple sprites. Broadcast & Wait allows for synchronization.

Repetitious Processing (looping) – repeated execution of code blocks to process large amounts of information or to control the repeated execution of code blocks required to direct the execution of a game or application. Use **Forever** & **Repeat** blocks.

Advanced Concepts:

Conditionals & Programming Logic - Use **If** & **If-Else** to check for a condition

Variables & Operators - Create your own variable (such as a score or timer) to keep track of numbers/number strings, generated by the use of Operators.

Boolean Logic – application of programming logic that executes based on the analysis of true/false data provided by Scratch during program execution.

Application & Game Development - Using all of the programming tools learned to create your own **Game**.