

Learner Full Name: _____ School: _____
Workstation Device: _____ Class: _____

Session 16: Block Programming & Algorithms

Until today, you have used software that other people built (Word, Excel, PowerPoint, Web Browsers). Today, you step across the line from being a technology *consumer* to becoming a technology **creator**: you will write your first computer program!

1. Plain Talk: The Kelewele Recipe (What is an Algorithm?)

Computers are lightning fast, but they are as dumb as a stone—they cannot guess what you want. They can only follow an **Algorithm**: a step-by-step list of clear, ordered instructions to accomplish a specific goal.

Think about making spicy fried plantain (*Kelewele*):



Figure 16.0: An Algorithm is a step-by-step sequence. The computer executes steps from left to right, top to bottom.

2. Why "Block-Based" Coding?

✗ TEXT-BASED CODE (Frustrating for Beginners)

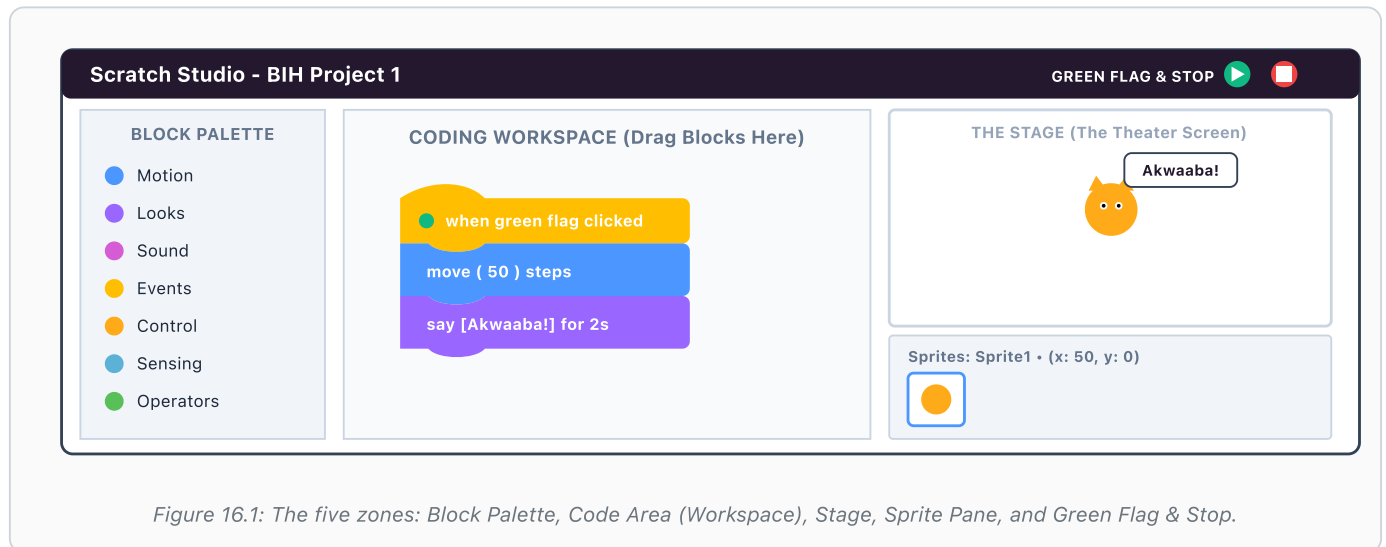
Typing code by hand in languages like Python or C++. If you forget a single semicolon (;) or comma, the entire program crashes with a confusing error message.

✓ BLOCK-BASED CODE (Visual & Fun!)

In **Scratch**, instructions are shaped like colorful **puzzle pieces**. You drag and snap them together. If two blocks don't make logical sense, they physically refuse to snap together!

1. The Five Major Zones of Scratch

When you launch **Scratch**, you are looking at a digital animation and game studio:



2. Essential Scratch Vocabulary

- **Sprite**: A character, person, animal, or object on the stage that you can control with code blocks.
- **Backdrop**: The background scenery picture behind your sprites on the stage (e.g. a classroom or football field).
- **Script**: A stack of two or more code blocks snapped together that run in order from top to bottom.
- **Green Flag**: The "GO" trigger button that starts your program.

1. Why Blocks Look Like Puzzle Pieces

Look closely at how Scratch blocks are built. They have **notches on the top** and **bumps on the bottom**. This guarantees that blocks connect only in ways that make logical sense to the computer:

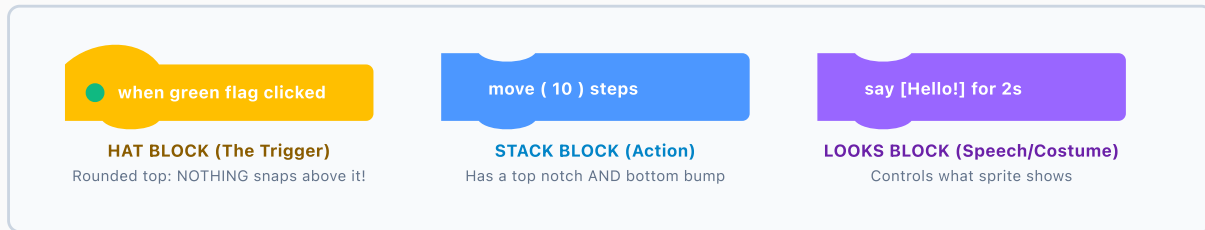


Figure 16.2: Anatomy of Scratch blocks. Hat blocks start scripts; Stack blocks carry out the actions.

2. The Golden Rule of Events: The Spark Plug

⚡ NO EVENT = NO ACTION!

If you snap together 10 motion blocks but forget to put a yellow **Event Hat Block** (like when green flag clicked) at the very top, the computer will sit completely still. The Event block is the electrical spark plug that ignites the code engine!

3. Core Motion Commands

Block Name	What It Does	Real-World Result
move (10) steps	Moves (jumps) the sprite forward in the direction it is facing.	Walking forward 10 tiny screen pixels.
turn clockwise (15) deg	Rotates the sprite to its right like turning a steering wheel.	Pivots right by 15 degrees.
say [Words] for (2) secs	Displays a cartoon speech bubble above the sprite's head.	Shows a speech bubble (no sound).



1. Where is My Sprite Standing?

To tell your character where to jump, walk, or appear on the stage, the computer uses a mathematical grid called the **X and Y Coordinate Plane**. This is just like finding a cell coordinate in a spreadsheet!

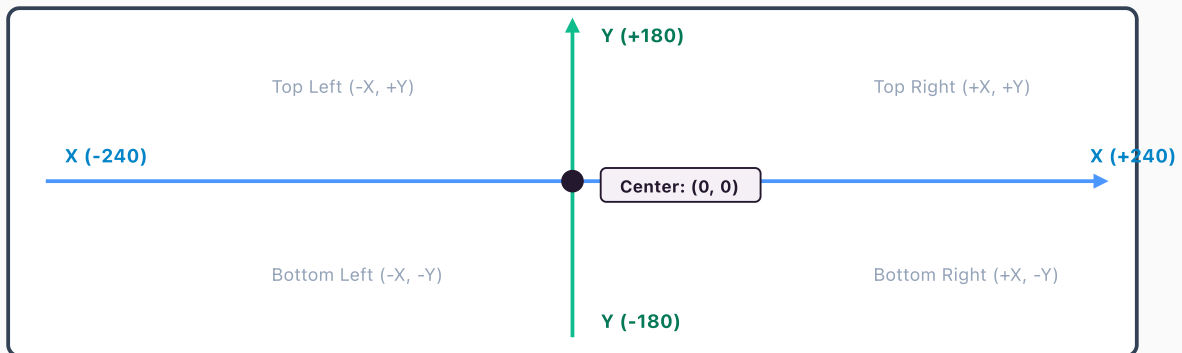


Figure 16.3: The Scratch Stage Grid. X controls horizontal movement (left/right); Y controls vertical height (up/down).

2. Coordinate Rules in Plain Language

X IS HORIZONTAL (Side to Side)

- Positive X numbers (+10, +100) move the sprite to the **RIGHT**.
- Negative X numbers (-10, -100) move the sprite to the **LEFT**.

Y IS VERTICAL (Up and Down)

- Positive Y numbers (+10, +100) push the sprite **UP** into the air.
- Negative Y numbers (-10, -100) drop the sprite **DOWN** to the ground.

3. Panic Button / Troubleshooting

MY SPRITE RAN OFF THE SCREEN!

If you tell a sprite to move 1,000 steps, it runs to the edge until only a sliver is left!

How to rescue it in 1 second:

1. In the Motion palette (blue), find go to x: () y: ().
2. Type 0 into both white circles, then click the block once.
3. Your sprite instantly snaps straight back to the dead center of the stage!



1. Lab Mission: Code an Animated Greeting Sprite

Follow these 6 steps to build your very first working Scratch program:

1

Launch Scratch:

Double-click the **Scratch Desktop** icon (or visit scratch.mit.edu in your browser).
You will see the famous Scratch Cat standing on the white stage.

2

Place the Event Spark:

Click on the yellow **Events** category dot on the left.
Drag the when green flag clicked hat block into the center Coding Workspace.

3

Make It Walk:

Click the blue **Motion** category.
Drag the move (10) steps block and snap it under the event block. Click inside the white circle and change 10 to 50.

4

Make It Speak:

Click the purple **Looks** category.
Drag the say [Hello!] for (2) seconds block and snap it directly below the motion block.
Click inside the white box and change "Hello!" to: Akwaaba! Welcome to BIH Tech Club!

5

Make It Turn:

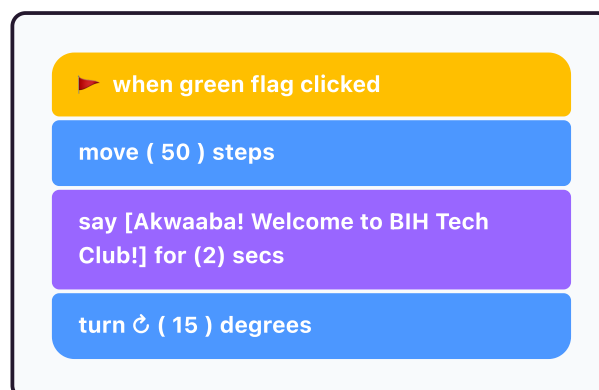
Go back to blue **Motion** → snap turn clockwise (15) degrees at the bottom of the stack.

6

TEST YOUR CODE! (The Big Moment):

Look at the top right of the screen. Click the bright **Green Flag**.
Watch: The cat walks forward 50 steps, shows its greeting bubble for 2 seconds, and pivots 15 degrees!
Save: **File** → **Save to your computer** → Name: Akwaaba_Cat_Ama.sb3.

2. Visual Target (Your Finished Code Stack)



1. Fix the Broken Algorithm Challenge

Below are 4 steps to wash a dirty school uniform, but they are scrambled in the wrong order! Write numbers 1, 2, 3, 4 to put the algorithm into the correct sequence:

Scrambled Task	Correct Step (1 to 4)
Hang the clean uniform on the drying line in the sun.	-----
Fill the washing basin with clean water and soap powder.	-----
Rinse the uniform in clean water to wash out the soap suds.	-----
Scrub the uniform gently to remove mud and dirt spots.	-----

2. Written Review Questions

Question 1: What is an **Algorithm** in computer science?

Question 2: What happens if you build a code stack in Scratch but forget to snap an **Event Hat Block** (like when green flag clicked) at the top?

Question 3: If you want a sprite to jump UP into the sky, do you increase its **X** coordinate or its **Y** coordinate?

3. Self-Assessment & Mentor Sign-Off

Tick the column that reflects your confidence at the end of Session 16:

Coding & Computational Thinking Competency	With Help	Almost Alone	All By Myself!
1. I can explain what an algorithm is and why instruction sequence matters.	☐	☐	☐
2. I can identify the 5 zones of Scratch (Palette, Workspace, Stage, Sprite Pane, Flag & Stop).	☐	☐	☐
3. I understand how X (horizontal) and Y (vertical) coordinates position sprites.	☐	☐	☐
4. I can assemble an Event, Motion, and Looks script and run it using the Green Flag.	☐	☐	☐

Mentor Signature: _____

Date Completed: _____