



CODING EDUCATION | 2026 PRACTITIONER EDITION

K-12 Coding Progression Framework for African Schools

A staged pathway from computational thinking and blocks to text-based programming, software design and independent problem-solving.

Jonathan Kennedy Sowah

AI Engineer and K-12 STEM Systems Builder
Ghana | Africa | jonathansowah.com

For curriculum teams, school leaders, teachers, coding providers and partners designing coherent K-12 pathways.

FIELD-TESTED THINKING | PRACTITIONER TEMPLATES | EVIDENCE ANCHORS

What this manual helps you decide

Move from idea to explicit decisions, owners, evidence and next actions.

1	Define what learning to code should mean	Coding education should develop reasoning, representation, creation, testing and explanation - not typing speed or memorised syntax.
2	Match tools to cognitive progression	Scratch, Python and JavaScript are not rival brands; they are environments with different affordances and transition costs.
3	Teach through projects without abandoning instruction	Projects create purpose, but learners still need modelling, deliberate practice, feedback and carefully managed cognitive load.
4	Design coding for ordinary African school conditions	A coherent pathway must work with shared devices, intermittent connectivity, varied language and uneven prior exposure.
5	Assess code as evidence of thinking	A running programme is evidence, but it is not enough to establish authorship, understanding, quality or transfer.
6	90-day application	Use a short design cycle to align capabilities, stages, teacher preparation, access and assessment before changing platforms.

Use the manual as a decision cycle

1 OBSERVE	2 DECIDE	3 ASSIGN	4 REVIEW
Start with the ordinary operating condition, not the launch-day exception.	Record the choice, the alternatives rejected and the assumption carrying the most risk.	Name one accountable owner, the supporting actors and the evidence they must produce.	Set a date and define what evidence will trigger continuation, repair, redesign or stop.

A completed template is not evidence of success. Pair every decision with an observable artefact, a named owner and a scheduled review.

Define what learning to code should mean

Coding education should develop reasoning, representation, creation, testing and explanation - not typing speed or memorised syntax.

A learner who can reproduce a known script has begun. A capable learner can decompose a problem, design a solution, test evidence and adapt in a new context.

DESIGN MOVE	WHAT IT REQUIRES	EVIDENCE TO KEEP
Reason	Decompose problems, recognise patterns, model state and choose useful abstractions.	Annotated plan and explanation
Create	Build an executable artefact for a user, purpose or authentic investigation.	Working programme and design rationale
Test	Predict behaviour, design cases, locate faults and justify corrections.	Test table, debug log and revision
Transfer	Apply concepts through an unfamiliar language, dataset, device or problem.	New-context performance task

Decision questions

1. What independent decision must the learner make?
2. What evidence shows understanding beyond a correct output?
3. Where does the pathway require explanation and critique?
4. Can the capability survive a change of software?

Decision gate

PROCEED WHEN	PAUSE OR REDESIGN WHEN
The programme names observable capabilities and authentic evidence at every stage.	The promise is limited to exposure, engagement, certificates or a list of languages.

Field application

Give learners a short unfamiliar programme containing a conceptual fault. Ask them to predict, test, repair and explain before teaching new syntax.

Match tools to cognitive progression

Scratch, Python and JavaScript are not rival brands; they are environments with different affordances and transition costs.

Choose a tool because it makes the intended concept learnable at the stage, then plan how learners will carry the concept into a less scaffolded environment.

DESIGN MOVE	WHAT IT REQUIRES	EVIDENCE TO KEEP
Early primary	Use sequencing, events, repetition, condition and representation through physical and block-based tasks.	Oral explanation and simple interactive artefact
Upper primary	Combine variables, procedures, input, debugging and purposeful project design.	Block project with planning and tests
Junior secondary	Bridge to text using equivalent concepts, trace tables, small programs and data.	Paired block-text task and debug evidence
Senior secondary	Develop modularity, algorithms, data structures, interfaces, collaboration and deployment awareness.	Independent software project and technical defence

Decision questions

1. Which concept does the environment reveal or hide?
2. What prerequisite is assumed at each transition?
3. How will learners see equivalence between block and text structures?
4. Does the sequence offer multiple meaningful destinations?

Decision gate

PROCEED WHEN	PAUSE OR REDESIGN WHEN
Tool changes follow demonstrated conceptual readiness and include explicit bridging.	Learners restart as beginners at every platform or are pushed into text before foundational reasoning is stable.

Field application

Select one concept such as iteration or state. Map how it appears in unplugged activity, blocks, Python and JavaScript, including common misconceptions.

Teach through projects without abandoning instruction

Projects create purpose, but learners still need modelling, deliberate practice, feedback and carefully managed cognitive load.

Strong coding pedagogy alternates explanation, worked examples, completion, modification, debugging and independent creation.

DESIGN MOVE	WHAT IT REQUIRES	EVIDENCE TO KEEP
Model	Make expert planning, tracing, naming, testing and debugging decisions visible.	Teacher think-aloud and worked example
Faded support	Move from completion to modification to independent construction as knowledge stabilises.	Task sequence showing reduced scaffolds
Feedback	Comment on reasoning, structure, tests and clarity - not only whether the code runs.	Feedback sample and learner revision
Collaboration	Use pair roles, code review and shared standards while preserving individual evidence.	Role record and individual explanation

Decision questions

1. What is directly taught before open project work?
2. Where do learners practise debugging deliberately?
3. How is feedback converted into a visible revision?
4. Can collaboration conceal unequal capability?

Decision gate

PROCEED WHEN	PAUSE OR REDESIGN WHEN
Instruction, practice and creation are deliberately sequenced and assessed.	Learners copy tutorials indefinitely or face open projects without the knowledge to make progress.

Field application

Observe a project lesson and count minutes spent deciding, typing, waiting, copying, testing and explaining. Redesign the largest low-value category.

Design coding for ordinary African school conditions

A coherent pathway must work with shared devices, intermittent connectivity, varied language and uneven prior exposure.

Constraint-aware design protects the learning objective through offline resources, structured grouping, accessible tasks and recoverable workflows.

DESIGN MOVE	WHAT IT REQUIRES	EVIDENCE TO KEEP
Offline core	Store editors, examples, documentation, datasets and submission routes locally where possible.	Offline lesson rehearsal and resource manifest
Shared devices	Use roles that rotate driver, navigator, tester and explainer while tracking individual decisions.	Participation observation and rotation record
Accessible representation	Combine text, diagrams, manipulatives, oral explanation and assistive options.	Accessible task variants and learner feedback
Entry points	Use baseline tasks and bridge modules rather than age-based assumptions about prior access.	Diagnostic profile and support pathway

Decision questions

1. What happens when internet access disappears?
2. Who touches the keyboard and who makes the technical decisions?
3. Which learners are penalised by language, device or typing assumptions?
4. Can work be recovered after power or device failure?

Decision gate

PROCEED WHEN	PAUSE OR REDESIGN WHEN
Every core objective has an accessible, tested delivery route under normal constraints.	Access depends on personal devices, continuous internet or informal help outside school.

Field application

Run the lesson at the worst credible learner-device ratio with the network disabled. Observe whether every learner still produces evidence of thinking.

Assess code as evidence of thinking

A running programme is evidence, but it is not enough to establish authorship, understanding, quality or transfer.

Use multiple evidence forms: planning, code, tests, debug history, explanation, review and performance in a changed context.

DESIGN MOVE	WHAT IT REQUIRES	EVIDENCE TO KEEP
Rubric	Judge problem interpretation, design, implementation, testing, debugging, explanation and responsible practice.	Shared rubric with exemplars
Authenticity	Use checkpoints, version history, oral defence and live modification instead of unreliable detection.	Process trail and individual demonstration
Moderation	Calibrate teacher judgement using common tasks and contrasting work samples.	Moderation record and adjusted standards
Progression	Track what learners do next: harder tasks, new contexts, peer support, study or work pathways.	Cohort progression and destination evidence

Decision questions

1. What capability does each assessment component reveal?
2. Can the learner modify the code and explain the consequence?
3. Would another teacher award a similar judgement?
4. Does the evidence distinguish support from substitution?

Decision gate

PROCEED WHEN	PAUSE OR REDESIGN WHEN
Assessment triangulates product, process, explanation and transfer.	Marks reward visual polish, length or correct output without evidence of reasoning.

Field application

Ask each learner to make one unprepared change to their own program, predict the effect and interpret the test result. Compare performance with submitted work.

Build a coherent coding pathway in 90 days

Use a short design cycle to align capabilities, stages, teacher preparation, access and assessment before changing platforms.

PERIOD	GATE	FIELD WORK	OUTPUT
Days 1-15	Diagnose	Observe ordinary conditions, establish the baseline and name the highest-risk assumptions.	Approved baseline and decision register
Days 16-30	Design	Select a bounded use case, define responsibilities and agree the evidence standard.	Signed design brief and risk controls
Days 31-60	Pilot	Run repeated practice under normal constraints; log failures, adaptations and participant experience.	Pilot evidence pack and repair log
Days 61-75	Review	Compare intended capability with observed performance and test equity, cost and continuity.	Review decision with conditions
Days 76-90	Institutionalise	Assign recurring ownership, budget, cadence and escalation routes before expansion.	Operating plan and next review date

Decision record

End-stage capability	_____
Learner baseline	_____
Priority concepts	_____
Tool and rationale	_____
Transition test	_____
Offline route	_____
Assessment evidence	_____
Next progression decision	_____

Do not ask which language is best in isolation. Ask which learning objective, learner stage, transition and operating condition the language must serve.

Evidence anchors and further reading

The frameworks translate public guidance and implementation evidence into practical decision tools.

SOURCE	RESOURCE	ADDRESS
Ghana Ministry of Education	Education Strategic Plan 2018-2030 (2018)	https://moe.gov.gh/edge/content/uploads/2019/05/Education-Strategic%20Plan-2018-2030.pdf
NaCCA	Secondary Education Curriculum (2025)	https://nacca.gov.gh/secondary-education-curriculum/
NaCCA	Robotics Curriculum for Secondary Education (2025)	https://nacca.gov.gh/wp-content/uploads/2025/04/ROBOTICS-Curriculum.pdf
African Union	Continental Education Strategy for Africa 2026-2035 (2025)	https://static.au.int/en/documents/20250616/continental-education-strategy-africa-2026-2035
World Bank	Devices for Education Knowledge Pack (2025)	https://documents1.worldbank.org/curated/en/099916207092523135/pdf/IDU-88a5de28-77c9-40f7-a841-9d425cdc505d.pdf

Use these sources as starting points. Confirm current law, policy, curriculum and regulator guidance before making consequential institutional decisions. This manual is an implementation aid, not legal, financial or safeguarding advice.

Evidence quality protocol

CHECK	QUESTION	RECORD
Currency	Is the source still current for the decision date, jurisdiction and learner group?	Version, publication date and date checked
Authority	Is it primary guidance, independent research, implementation evidence or vendor material?	Source type and responsible institution
Fit	Do its assumptions match the school, infrastructure, language, staffing and safeguarding context?	Transfer limits and local verification needed
Use	Which decision does this source inform, and what would cause that decision to change?	Decision link, owner and review trigger

Move learners from syntax exposure to software capability

A practical contribution to stronger AI, coding, robotics and STEM education systems in Ghana and Africa.

The value of a framework is not the elegance of its diagram. It is the quality of the decisions, routines and learning that it makes possible.

About Jonathan Kennedy Sowah

Jonathan is a Ghanaian AI engineer and K-12 STEM systems builder working across robotics, coding, applied artificial intelligence, teacher development and context-aware education technology. His wider portfolio has reached 25,000+ learners, trained 2,500+ educators and engaged 100+ schools. KodeVR has more than 10,000 active learners.

He works with schools, governments, companies and development partners to turn high-visibility technology ambitions into practical learning systems that can operate under real African conditions.

Use this work responsibly

Adapt the tools to the programme, participants, institution and regulatory context. Record assumptions. Seek qualified technical, legal, safeguarding, financial and evaluation review where the decision requires it.

Before commissioning implementation support

QUESTION	A STRONG ANSWER SHOULD NAME
What capability must remain after the engagement?	The learner, teacher and institutional behaviours that should continue without external supervision.
What conditions shape delivery?	The ordinary timetable, infrastructure, language, safeguarding, procurement and maintenance realities.
How will quality be governed?	Decision owners, evidence definitions, review cadence, repair route and stop-or-scale thresholds.
What will the institution inherit?	Reusable learning assets, local capability, documented routines, data rights and an affordable continuity plan.

jonathansowah.com | info@jonathansowah.com