PADMA ROBOTICS PROGRAMME — CURRICULUM MAP

Terms 1–3 · Lesson by lesson · Mapped to Cambridge Lower Secondary and Pearson Edexcel iLowerSecondary Computing

Draft v0.2 · For verification · Awaken Innovation Lab

⚠ Read this before the map goes to a client

This document is built on **competence statements, not official descriptor codes.**

I have not verified the current Pearson Edexcel iLowerSecondary Computing specification, nor the current Cambridge Lower Secondary Computing curriculum framework. Both boards have restructured their international qualifications in recent years, and the exact strand names, descriptor codes and assessment components available to a private training centre must be confirmed directly with the board before any of this is quoted to a parent.

The design of this map protects you from that uncertainty. Every descriptor below sits in its own column with a stable code of our own (`CT1`, `PD3`, …). When the official specifications are confirmed:

- replace the **official code** in the left-hand column for the board you are quoting,
- leave the statement, the lesson mapping and the evidence column untouched.

Nothing else in this document changes. That is deliberate — a map that has to be rebuilt every time a specification is updated is not a map, it is a liability.

One document, two boards. A school following Cambridge can hand this map to a parent on day one, and a school following Pearson can do the same.

STATUS	ITEM
✅ Verified	Lesson sequence, evidence artefacts, internal descriptor set
⬜ **Unverified**	Pearson strand names, descriptor codes, assessment availability
⬜ **Unverified**	Whether iLowerSecondary Computing is currently offered to private centres in your region
⬜ **Unverified**	Cambridge strand names used in §8; confirm against the current Cambridge Lower Secondary Computing framework

STATUS	ITEM
⬜ **Unverified**	Whether Cambridge Lower Secondary Computing is offered to private centres in your region, and on what terms

---

1. What this map claims, and what it does not

It claims: after three terms, a student can evidence secure competence in computational thinking, programming constructs, and physical computing at the standard expected of an 11–14 computing curriculum.

It does not claim: that a student has achieved a Pearson or Cambridge qualification, sits at iGCSE level, or can be entered for an external examination on the strength of this course.

The difference matters commercially. A training centre that resells an overstated claim owns the reputational damage, and so do you.

---

2. Descriptor set (working codes)

Twenty-seven competences, in five strands. Written to describe what a student *can do*, not what they have *heard of* — so each can be evidenced by an artefact.

Computational thinking (CT)

CODE	STATEMENT
CT1	Decompose a problem into parts small enough to build and test separately
CT2	Design a sequence of precise instructions that produces a specified outcome
CT3	Recognise repetition and pattern in a problem and represent it compactly
CT4	Use abstraction — a name, a block, a table — to hide detail that is not needed
CT5	Evaluate a solution against stated criteria and refine it
CT6	Reason about cause and effect in a system: what a change will do before running it

Programming and development (PD)

CODE	STATEMENT
PD1	Construct a program using sequence, with correct order and parameters
PD2	Use events to control program flow
PD3	Use iteration (counted and conditional loops)
PD4	Use selection, including nested and compound conditions
PD5	Declare, update and display variables
PD6	Create and call functions, including with parameters
PD7	Test systematically and debug using an isolation method
PD8	Document a program so another person can use or rebuild it
PD9	Create a list, add to it, read by position, and use it to hold a record

Notation (CF)

CODE	STATEMENT
CF1	Draw a flowchart representing a sequence, using the standard symbols
CF2	Draw a flowchart containing a decision
CF3	Draw a flowchart containing a loop, using a back-arrow

Data and systems (DS)

CODE	STATEMENT
DS1	Identify input, process and output in a system
DS2	Use sensors to capture data about the physical world
DS3	Control an actuator from a program
DS4	Calibrate a system and justify threshold values
DS5	Explain and implement a feedback control loop
DS6	Diagnose a fault across code, hardware, power and environment

Communication (CM)

CODE	STATEMENT
CM1	Explain a design decision to a listener in their own words
CM2	Work in defined roles and hand over responsibility
CM3	Produce documentation for a named audience

Level markers used in the tables: ○ introduced · ◐ practised · ● assessed (charged to the term grade)

---

3. Programme assumptions

This map describes the **blended programme** agreed in design review, not the earlier standalone editions.

ASSUMPTION	VALUE
Structure	3 terms × 12 weeks (10 teaching + sprint + showcase)
Contact time	2 × 40 min per week, ~54 hours per year
Grades	6–8 (ages 11–14)
Hardware	KanKaung line-tracker, ESP32
Firmware watchdog	Present — motors stop if commands cease
Robot in Term 1–2	Remote control only, **rotating: half the tables per week**
Term 3	Assumes Term 2 — conditionals and variables are not re-taught
Notation	Flowcharts from Term 1, reused through Term 3 and carried into Bridge
Object orientation	**Not taught.** Padma has no class construct; it belongs in Python
8-pin expansion	Club only, outside the assessed curriculum
Assessment	Four pillars: Logic · Cleanliness · Debugging · Collaboration

> **On the descriptor count.** This map has **27 descriptors**. The v0.2 revision added the three notation descriptors (CF1–CF3) and the list descriptor (PD9), and made causal reasoning (CT6) and fault diagnosis (DS6) explicit — because the programme was teaching flowcharts, lists and hardware faults without assessing them. A descriptor that is taught but never assessed is invisible to a moderator, and therefore not evidence.

---

4. TERM 1 — SIGNALS

"You drive it."

WK	LESSON	NEW CONCEPT	DESCRIPTORS	EVIDENCE ARTEFACT
1	Meet Padma, Meet the Robot	Program, instruction, hat block	PD1 ○ · PD2 ○ · CT1 ○ · CM1 ○	First script saved as `P1_01`; notebook p.1
2	Precision Drawing	Sequence, parameters, angles — **first flowchart**	PD1 ◐ · CT2 ○ · CT3 ○ · **CF1** ●	Printed square + triangle; **flowchart of the square, hand-drawn**; one written explanation of the 120° turn
3	The Stage as a Map	Coordinates, heading	CT2 ◐ · CT4 ○ · PD1 ◐	Route drawn on paper, then built; end position within tolerance
4	Mission Control	Events, broadcast	PD2 ● · CT6 ○ · DS1 ○ · DS3 ○ · CM2 ◐	**Four-sprite remote drives KanKaung** (rotating half). Video, 20 s. Board photo of the four buttons.
5	Start, Stop, State	State machine, costumes — **loop-back arrow**	PD2 ● · CT6 ◐ · PD5 ○ · **CF3** ●	Three-state patrol; state diagram **and flowchart** copied into notebook
6	Loops	`repeat N`, refactoring	PD3 ○ · CT3 ●	Refactor before/after: 8 blocks → 2. Block counts recorded.
7	Nested Loops	Loop inside a loop	PD3 ◐ · CT3 ● · CT4 ◐	**Lotus tile wall**, printed; outer/inner loop sentence
8	Composition	Combining three constructs	CT1 ◐ · CT5 ○ · PD1–PD3 ◐	Reaction patrol running 60 s unattended
9	Debugging Studio	Six-step protocol	PD7 ○ · CT5 ◐	Six-row bug log, teacher-initialled
10	Build Sprint	Decomposition, planning	CT1 ● · PD7 ◐ · PD8 ○ · CM3 ○	Signed planning sheet + working console + operator card

Term 1 assesses: PD1, PD2, PD3, CT3, CT1, **CF1, CF3 Term 1 introduces but does not charge:** DS1, DS3 (through the Week 4 robot rotation)

---

5. TERM 2 — RULES

"You teach it to decide."

WK	LESSON	NEW CONCEPT	DESCRIPTORS	EVIDENCE ARTEFACT
1	The Decision Problem	Conditions, true/false, measurable facts, **AND / OR / NOT gates**	PD4 ○ · CT6 ○	Sorted statement cards; **four-row AND and OR truth tables**
2	If / Else	Branches, comparison operators	PD4 ◐ · CT2 ◐	Airlock project + five-row test table
3	Compound Conditions	`and` / `or` / `not` — **table → flowchart**	PD4 ◐ · CT2 ◐ · CT4 ○ · **CF2 ●**	Decision table **and four-state flowchart**, hand-drawn, teacher-initialled
4	Variables	Set, change, display	PD5 ○ · CT4 ◐	Power grid with clamping; negative-value bug fixed
5	Variables That Matter	Parameters, calibration	PD5 ● · DS4 ○ · PD7 ◐	**Trim bench:** robot driven straight (rotating half). Five-run drift table.
6	Modes	One variable drives all behaviour	PD5 ● · PD4 ◐ · CT6 ●	Four-mode state diagram + working implementation
7	Functions	Custom blocks	PD6 ○ · CT4 ●	Duplicated program refactored; before/after block count
8	Parameters	Generalising a routine — **the radar sweep**	PD6 ● · CT1 ◐	`scan(angle)` sweeping 0–180; the arc visible on the stage
9	Integration	Contracts, message bus	CT5 ● · CM2 ● · PD8 ◐	Interface contract sheet, both teams signed
10	Testing	Boundary values, hostile input	PD7 ● · CT5 ● · PD8 ◐	Eight-row test table including boundaries; bug reports

Term 2 assesses: PD4, PD5, PD6, PD7, CT5, CT6, CM2, **CF2 Term 2 prepares Term 3:** the four-state table **and its flowchart** are written on paper here, and carried forward.

---

6. TERM 3 — AUTONOMY

"It drives itself."

WK	LESSON	NEW CONCEPT	DESCRIPTORS	EVIDENCE ARTEFACT
1	Meet KanKaung	GPIO, pin map, first motor command	DS1 ● · DS3 ● · DS6 ○	Robot drives forward and stops on command; labelled pin map
2	Trim, Properly	Why trim drifts: battery sag, heat, wear	DS4 ● · DS6 ◐ · CT6 ◐	Three trim runs at three battery levels; written explanation of the drift
3	What the Sensors See	IR reflectance, both sensors read	DS2 ○ · CT6 ◐	Physical truth table, robot moved by hand, motors off
4	Four States, Four Branches	Selection applied to a real machine	PD4 ● · PD3 ◐ · CT6 ●	**Their own Term 2 table, now running.** Video of a straight-line follow.
5	Variables and Loop Rate	Speed, correction, clamping, screen refresh	PD5 ● · PD3 ● · PD7 ◐	Three-speed timing table; oscillation identified and named
6	Tuning	Speed vs stability, overshoot, damping	CT5 ● · CT6 ●	Speed/correction/success-rate table; one-sentence justification
7	The Second Sense	Ultrasonic distance, thresholds, hysteresis	DS2 ● · DS4 ● · DS5 ○	Deadband worksheet; 15-second no-flicker demonstration
8	Priority and Fail-Safe	Override > obstacle > line; the "unknown" state	DS5 ● · DS6 ● · CT1 ●	Priority ladder written from memory; unplugged-sensor test
9	Debugging a Physical System	Four fault categories, the **black box list**	DS6 ● · PD7 ● · **PD9** ●	`faults` **list** on stage, read aloud; five injected faults categorised
10	Build Sprint	Course completion, reliability	CT1 ● · CT5 ● · PD8 ● · CM3 ●	**Course completed 8/10.** Lap times. Service card.

Term 3 assesses: DS1–DS6, PD3–PD9, CT1, CT5, CT6, CM3, **PD9**

---

7. Coverage matrix — where each descriptor is secured

DESCRIPTOR	T1	T2	T3	SECURED BY
CT1 Decompose	○	◐	●	T3 W10 build sprint

DESCRIPTOR	T1	T2	T3	SECURED BY
CT2 Algorithm design	○	◐	◐	T1 W3 route planning
CT3 Pattern & repetition	●	—	—	T1 W6–W7 loops
CT4 Abstraction	○	●	◐	T2 W7 functions
CT5 Evaluation & refinement	◐	●	●	T3 W10
CT6 Causal reasoning	○	●	●	T3 W6 tuning
PD1 Sequence	●	—	◐	T1 W2
PD2 Events	●	—	—	T1 W4
PD3 Iteration	●	—	●	T1 W6, T3 W5
PD4 Selection	—	●	●	T2 W1–W3, T3 W4
PD5 Variables	○	●	●	T2 W5, T3 W5
PD6 Functions	—	●	◐	T2 W7–W8
PD7 Testing & debugging	◐	●	●	T2 W10, T3 W9
PD8 Documentation	○	◐	●	T3 W10 service card
PD9 Lists	—	—	●	**T3 W9 black box**
CF1 Sequence flowchart	●	—	—	**T1 W2**
CF2 Decision flowchart	—	●	—	**T2 W3**
CF3 Loop flowchart	●	—	◐	**T1 W5**
DS1 Input–process–output	○	—	●	T3 W1
DS2 Sensors & data	—	—	●	T3 W3, W7
DS3 Actuator control	○	—	●	T3 W1
DS4 Calibration & thresholds	—	○	●	T3 W2, W7
DS5 Feedback loops	—	—	●	T3 W7–W8
DS6 Fault diagnosis	—	—	●	T3 W9
CM1 Explanation	●	◐	◐	T1 W10 defence
CM2 Collaboration	◐	●	◐	T2 W9 integration

DESCRIPTOR	T1	T2	T3	SECURED BY
CM3 Audience documentation	○	◐	●	T3 W10 service card

Reading the matrix. ○ introduced, ◐ developing, ● secured. Every descriptor reaches ● by the end of Term 3 — none is taught and never checked. Where a descriptor shows ● in two terms, the later term is the one to cite on a transcript; the earlier is the first secure judgement.

---

8. Alignment — Cambridge

Strand-level only. Cambridge publishes strands, not the internal codes used here, so each row names the strand the descriptor sits in. Confirm the strand names against the current framework before quoting them. No Cambridge descriptor code, component number or assessment claim is made in this section.

CODE	DESCRIPTOR	CAMBRIDGE STRAND
CT1	Decompose	Computational Thinking
CT2	Algorithm design	Computational Thinking
CT3	Pattern and repetition	Computational Thinking
CT4	Abstraction	Computational Thinking
CT5	Evaluation and refinement	Computational Thinking
CT6	Causal reasoning	Computational Thinking
PD1	Sequence	Programming
PD2	Events	Programming
PD3	Iteration	Programming
PD4	Selection	Programming
PD5	Variables	Programming
PD6	Functions	Programming
PD7	Testing and debugging	Programming
PD8	Documentation	Programming
PD9	Lists	Managing Data
CF1	Sequence flowchart	Computational Thinking

CODE	DESCRIPTOR	CAMBRIDGE STRAND
CF2	Decision flowchart	Computational Thinking
CF3	Loop flowchart	Computational Thinking
DS1	Input–process–output	Computer Systems
DS2	Sensors and data	Managing Data
DS3	Actuator control	Computer Systems
DS4	Calibration and thresholds	Computer Systems
DS5	Feedback loops	Computer Systems
DS6	Fault diagnosis	Computer Systems
CM1	Explanation	Networks and Digital Communication
CM2	Collaboration	Networks and Digital Communication
CM3	Audience documentation	Networks and Digital Communication

Where the fit is loose. The communication descriptors (CM1–CM3) are speaking, working and writing, not networking. They are placed in Networks and Digital Communication because that is the closest Cambridge strand; treat those three rows as an approximation and say so if a parent asks.

---

9. The achievement statement

The sentence a training centre may quote, verbatim:

> *"After three terms, students can evidence secure competence in computational thinking, programming and physical computing at the standard expected of an 11–14 computing curriculum. They will have designed, built and tested a working line-following robot, and can explain and document their own design. Students are ready to progress to a text-based programming course preparing for iGCSE Computer Science."*

Three things this does deliberately:

1. **It names the age band, not a grade.** "Standard expected of an 11–14 curriculum" is defensible.
2. **It points to the artefact.** A robot that follows a line is evidence a parent can watch.
3. **It names the next step.** A programme that says what comes after is trusted more than one that claims completion.

What it does not say: *Level 2*, *iGCSE standard*, *equivalent to*. Those words invite a comparison you will lose.

---

10. Evidence pack — what a moderator would actually read

Assembled per student, per year. Roughly one folder.

ITEM	FROM	PURPOSE
Engineering notebook	All terms	The only continuous record of thinking
Six-row bug log	T1 W9	PD7 evidence
Decision table	T2 W3	PD4 evidence
Test table with boundaries	T2 W10	PD7 evidence
Four-state table (paper)	T2 W3	The document that later runs
Trim drift table	T3 W2	DS4 evidence
Fault log, five faults	T3 W9	DS6 evidence
Service card	T3 W10	PD8 + CM3 evidence
90-second course video	T3 W10	The claim, made visible
Operator card	T1 W10	CM3 emerging

Two of these are worth filming. A parent will not read a test table, but they will watch a robot complete a course.

---

11. Verification checklist — before this goes to a client

- [ ] Confirm **Cambridge Lower Secondary Computing** is currently offered to your centre, its exact name, and which curriculum framework and strands apply.
- [ ] Confirm the **five Cambridge strand names** used in §8 — Computational Thinking, Programming, Managing Data, Networks and Digital Communication, Computer Systems — against the current framework.
- [ ] Confirm whether a **private training centre** can enter candidates for Cambridge Lower Secondary Checkpoint, or whether it is school-only.
- [ ] Confirm **iLowerSecondary Computing** is currently offered, its exact name, and which strands it uses.

- [ ] Confirm whether a **private training centre** can enter candidates for any iLowerSecondary assessment, or whether it is school-only.
- [ ] Obtain the official **strand and descriptor list** from each board; replace the `CT/PD/CF/DS/CM` codes with the board's own wording.
- [ ] Confirm **iGCSE Computer Science** entry requirements for private candidates in your region — some centres cannot enter private candidates for the practical component.
- [ ] Check whether Pearson offers a **progress test** rather than a qualification; a progress test may be the honest, sellable option.
- [ ] Legal/branding check: the words "Pearson" and "Cambridge" cannot appear in marketing without permission. Use "*mapped to*", never "*accredited by*".

---

12. What changes in the lesson packs

This map is the target state. Two packs move to meet it:

PACK	CHANGE
Term 1	Week 4 and Week 6 gain the rotating robot activity. Week 8 loses "First Contact" and becomes **Composition**.
Term 2	Week 5 gains the trim bench; Week 8 gains the countdown-on-robot. Week 3 gains the written four-state table carried into Term 3.
Term 3	Rewritten to assume Term 2. Week 2 becomes "*why trim drifts*", not "*what trim is*". GPIO expansion moves to the club.

---

13. What this document does not solve

It does not make the course a qualification. No amount of mapping does that. If the client needs a certificate with an awarding-body logo on it, that is a different commercial conversation with Pearson or Cambridge, and it takes months, not weeks.

It does not cover Term 4. If a Bridge to Text year is built, it needs its own map against iGCSE Computer Science — which is a genuinely different descriptor set, with text-based programming and formal theory.

It does not replace moderation. A map says what *should* be evidenced. Only a second pair of eyes on real student work confirms that it *is*.

Padma Robotics Programme curriculum map v0.2 · Awaken Innovation Lab · Draft — Pearson and Cambridge descriptors unverified pending confirmation.