[New] Detailed PEAR Programme

Programme Name: Program for Excellence in African Research (PEAR)
Dates: Monday 18 August – Wednesday 27 August 2025
Venue: Greensprings School, Anthony Campus, Lagos, Nigeria
Schedule: Daily from 09:00 to 17:00 excluding Sunday. Office hours from 16:00 to 17:00
Participants: 30 high school students, teams of 3
Format: Day programme (commuter-based), lunch provided for all students; accommodation and other meals provided for select students with need
Teaching Team: 2 Lead Facilitators (Emmanuela Ilok & Tony Odhiambo) + 3 mentors
Tuition: No Tuition/Programme fees paid by the students.

Programme Structure and Learning Outcomes

The PEAR Programme delivers an immersive, hands-on STEM experience grounded in real-world problem solving.

Students progress through a blend of guided workshops, structured hardware/software builds, and interdisciplinary sessions that emphasise both technical mastery and communication.

Each day is designed to follow a natural rhythm: beginning with light engagement and warm-up activities, moving into focused technical work, and concluding with collaborative reflection, feedback, and presentation practice.

Core Learning Pillars

1. Software Development and Data Interpretation

- Focus on writing clear and effective Python code for real-world tasks
- Students explore key programming constructs and apply them to practical data tasks
- Use of `pandas` and `matplotlib` to clean, analyse, and visualise simple datasets
- Emphasis on drawing insights from data and clearly communicating results
- All work is completed in Google Colab to ensure accessibility and ease of feedback

2. Hardware and IoT Systems

- Students build tabletop-scale systems using the **SunFounder Pico 2W kit**

- Projects involve real-world sensing (e.g. soil moisture, temperature, light levels) and response logic (e.g. LED, buzzer alerts)
- Each build is framed around a practical challenge such as environmental monitoring or hygiene
- Systems are developed using MicroPython, with support from structured circuit diagrams and step-by-step guides
- Students collect, log, and interpret sensor data as part of their learning

3. Interdisciplinary Thinking and Scientific Communication

- Capstone projects integrate software, hardware, and research thinking
- Students use the scientific method to test hypotheses, design experiments, and refine ideas
- Communication sessions support poster creation, one-page summaries, and oral presentations
- Emphasis on explaining STEM ideas clearly to both technical and non-technical audiences
- Teams receive feedback throughout the programme and refine their messaging accordingly

Primary Learning Outcomes

By the end of the program, students will be able to:

- Write and debug Python code for real-world tasks
- Use hardware components to sense and respond to environmental conditions
- Interpret data, identify patterns, and draw evidence-based conclusions
- Collaborate in teams to plan, build, test, and present a working STEM prototype
- Communicate findings through visual aids, written summaries, and presentations

Before the Programme Starts:

- Accepted students will receive a welcome + congratulations email with further program information
- They will fill out a short survey to assess their familiarity with Python and electronics.
- They will also fill consent forms for media release (photos/videos/social media)
- Included will be a light pre-programme task (e.g. some readings and videos on Python/microcontrollers as an introduction)
- Their responses will be used to shape mentor support and session pacing

Student Welcome Packet

Upon arrival, each participant will receive a personalised PEAR welcome kit containing:

- **PEAR-branded tote bag** – to carry daily materials and components
- **PEAR notebook** – for notes, sketches, circuit diagrams, and reflections
- **PEAR t-shirt** – to foster team spirit and for use during demo day
- **Programme schedule printout** – outlining daily activities and sessions
- **Name badge and lanyard** – for easy team and mentor identification
- **Basic stationery** – for quick annotations
- **Folder** - to store handouts (will be available both online + hardcopy)

Typical Daily Schedule

09:00 – 11:00 | Morning Warm-Up and Preparation

- Light-touch activities to ease into the day and stimulate creative thinking
- Includes: Ice-breakers and energisers, Foundational coding challenges, Design-thinking exercises, Mentor check-ins and team sync-ups
- Purpose: build team rapport, reinforce key concepts, and mentally prepare students for focused technical work

11:00 – 15:00 | Core Technical Workshops (with Lunch Break)

- Dedicated time for intensive learning and hands-on project development
- Students work on either software or hardware tracks, depending on the day's module
- Activities include:
    - Python programming and data analysis in Colab
    - Sensor wiring, circuit prototyping, and system testing
    - Guided hardware builds or capstone development
- Lunch is integrated into this block
- Mentors are fully available during this period to provide in-depth support and guidance

15:00 – 16:00 | Capstone, Applied Communication & Reflection

- A shift to lighter but purposeful tasks focused on communication and synthesis
- Activities may include:

    - Capstone project planning and iteration
    - Poster design and visual storytelling exercises
    - Peer feedback and group critique
    - Oral presentation drills or games
- Focus is placed on reinforcing learning through articulation and constructive discussion

16:00 – 17:00 | Optional Office Hours

- Unstructured time for additional support and enrichment
- Students may choose to:
    - Troubleshoot technical challenges
    - Seek personalised feedback on code, circuit logic, or presentations
    - Consult mentors on academic or career advice

Day-by-Day Plan

Monday 18 Aug – Arrival & Orientation.

- Orientation session and welcome ceremony to introduce students to the Programme and its objectives
- Introductory demonstration on the practical relevance of Python and the Raspberry Pi Pico
- Light, exploratory activities to help students familiarise themselves with the tools and environment
- Initial engagement with mentors and peers to begin identifying real-world problems of interest

Tuesday 19 – Software Sprints I: Python Fundamentals

- Students begin structured review of core Python concepts: variables, loops, conditionals, functions, and lists
- Pair-programming exercises emphasise problem-solving and debugging skills
- Real-world mini-tasks using CSVs and pandas: e.g., calculating average rainfall, filtering data by condition
- matplotlib intro: students create simple plots to visualise cleaned data
- Mentors circulate to provide code review, suggest improvements, and address conceptual gaps
- Afternoon mini-challenge: each group interprets a dataset and presents one chart + one insight in 2 minutes
- **Capstone Session 1:**

- Programme leads introduce capstone goals and example themes (e.g. food systems, environment, urban safety)
- Students form project teams and begin exploring which data or system they may want to work with
- Each group completes a brief capstone team charter (members, shared interest, initial problem ideas)

Wednesday 20 – Software Sprints II: Data Meets Hardware

- Students work with mock sensor datasets (temperature, light, soil moisture) formatted as CSV files
- In Python, teams clean, filter, and visualise these datasets using pandas and matplotlib
- Guided prompts push students to draw meaningful conclusions from patterns in the data
- Example: "What time of day does the soil dry fastest?" or "What does this spike suggest?"
- Mentors help students articulate both what their code does and what their data shows
- Teams each prepare and share a short 2-slide deck (1 graph + 1 key insight) with peer questions encouraged
- **Capstone Session 2:**
    - Teams finalise a clear, concise problem statement for their capstone
    - Identify physical components they may use and the data needed for feedback or automation
    - Begin sketching input–processing–output relationships in plain English

Thursday 21 – Hardware Foundations + Build Day 1

- Students are introduced to key Pico kit components: breadboard, resistors, sensors, buttons, buzzer, LCD, and servo
- Hands-on mini-labs: blink an LED, read temperature from a sensor, and output values to serial or LCD
- MicroPython exercises cover reading sensor inputs, setting basic threshold logic, and displaying live output data.
- Mentors support students to improve wiring reliability and understand signal variability and unit precision
- **Build 1: Digital Cashier Terminal**
    - Teams build a simple digital point-of-sale terminal using only kit components

  - Use keypad (or buttons) to select an item from a predefined menu and input quantity
  - LCD shows selected item, quantity, and running total
  - Optional : buzzer/LED for confirmation
  - Teams test each other's cashier setups in a mini-mock tuckshop scenario
- **Capstone Session 3:**
  - Teams sketch a basic system diagram of their proposed build: inputs, decision logic, and outputs
  - Draft a minimal pseudocode to represent their idea's data or control flow
  - Mentor reviews and feedback shared to shape next steps

Friday 22 – Build Day 2: Smart Plant Guardian

- Guided build focused on using soil moisture and temperature sensors to monitor plant health
- Teams wire sensors to the Pico, configure digital/analog pins, and set LED indicators for dry vs. healthy conditions
- Students implement a basic logging routine and begin a timed test (e.g., record moisture every 3 minutes for 1 hour)
- During review, students graph moisture vs. time, and compare changes across teams with different test conditions (e.g., sunlight, watering)
- Mentors facilitate brief discussion on how to tune thresholds and interpret environmental influence
- **Capstone Session 4:**
  - Teams apply lessons from the build to inform their own capstone prototype plan
  - Begin listing specific data points their system should collect and start sketching their circuit layout
  - Each team writes a one-paragraph "design statement" and shares it with mentors for feedback

Saturday 23 – African Innovation & Design Lab + Capstone Integration

Morning Session: African Innovation & Design Lab (09:00–12:00)

- An interactive, story-driven session focused on real African innovations in STEM, followed by a guided design challenge.
- Part 1 – **Inspiration through African Case Studies (Facilitator-led)**
  - Students are introduced to 2–3 brief, powerful case studies of African-led innovations
  - Examples may include:
    - ❖ M-KOPA: solar home systems with pay-as-you-go financing

        - ❖ Ubenwa: using baby cries to diagnose birth asphyxia with AI
        - ❖ Zipline: drone delivery of blood and vaccines in remote areas
        - ❖ Wecyclers: community-powered recycling for urban waste management
        - ❖ FarmDrive: digitising farmer records for access to micro-loans
- Each story should highlight:
    - The problem (local, real, pressing)
    - The technical or design solution (even if basic tech)
    - The impact on the community
- Part 2 – **“Local Lens” Innovation Sprint**
    - Students then break into new small groups (not capstone teams) and are given one challenge prompt based on local issues (e.g. water waste, urban transport, classroom heat, power instability)
    - Using what they’ve learned from the case studies, each group has 1 hour to:
        - ❖ Define a localised version of the problem
        - ❖ Propose a low-tech or tech-enabled concept solution
        - ❖ Sketch or storyboard how it would work
        - ❖ Prepare a 3-minute verbal pitch + 1-slide visual aid
    - Mentors support the groups as roaming facilitators - pushing critical thinking, encouraging feasibility checks, and refining storytelling.
- Why this works:
    - Directly builds design thinking and real-world relevance
    - Gets students excited by African problem-solvers
    - Prepares them to frame and pitch their own capstones better
    - Develops problem-context-awareness, a huge gap in many STEM competitions

Capstone Session 5: Iteration and Planning

- Students return to their capstone teams
- Reflect on the design techniques used earlier
- Refine their capstone’s value proposition, communication strategy, and system logic
- Begin outlining slides or posters for demo day
- Mentors provide targeted coaching on clarity and impact

Sunday 24 – Rest

- No formal sessions are scheduled
- Residential students may remain on campus and engage in informal, self-paced activities as needed

- Commuting students have the day off
- Mentors and facilitators may use this time to update project materials, review student progress, and prepare for the final phase of the programme

Monday 25 – Build Day 3: Auto-Dimming Temperature Display

- Students build a system that displays changes in temperature using an LED bar and adjusts LED brightness automatically based on ambient light.
- The thermistor is calibrated to room temperature at startup, so the LED bar shows how much hotter or cooler the environment becomes over time.
- A photoresistor controls the overall brightness of the LEDs, mimicking how phones adapt their screens to different lighting conditions.
- Students use rolling averages to smooth sensor readings and avoid flickering or overreacting to sudden changes.
- Students reflect on how ambient conditions affect visibility, and how smart lighting systems could improve user comfort and energy efficiency in real devices.
- **Capstone Session 6:**
    - Capstone teams implement or test key hardware/software components
    - Each group begins live logging or test-run scripts to generate first capstone data
    - Students write a short technical note: "What one thing failed today and how we fixed it"

Tuesday 26 – Build Day 4: Contact-Free Hand-Wash Timer

- Final guided build: students use an ultrasonic sensor to detect hand presence and initiate a 20-second LED countdown
- Circuit includes visual and/or audio indicators (e.g., buzzer at end of timer)
- Teams test under real-use scenarios: how often do false triggers occur? Do users wait for the full duration?
- Students collect usage data across short intervals and analyse compliance trends
- Mentors prompt groups to consider "design for behaviour" - how hardware nudges real action
- **Capstone Session 7:**
    - Full prototype testing and integration: students complete capstone builds and begin structured functionality testing
    - Each group drafts the content for their final communication materials (poster or slides):
        - Problem & context
        - System diagram
        - Data collected

        - ❖ Insights gained
        - ❖ Impact or recommendation
    - Mentors conduct mini Q&A rounds to simulate Demo Day and help refine explanations

Wednesday 27 – Final Presentations & Closing

- Mentors and facilitators should prepare a **formal invite list** in advance, including local teachers, tech professionals, university faculty (UNILAG, Covenant etc), and parents. A Google Form RSVP may be used to manage attendance.
- 09:00–11:00:
    - Final checks, demo table setup, poster printing, and group rehearsals
    - Mentors circulate to run last-minute feedback or troubleshoot issues
- 11:00–15:00 – **PEAR Demo Day**:
    - Each capstone team presents in a structured 6-minute format followed by 3 minutes of Q&A
    - Presentations must explain the problem, demonstrate system function, present collected data, and share conclusions
    - Panels include mentors and invited educators or STEM professionals
    - Judges assess clarity, technical execution, innovation, relevance, and communication
- 15:00–17:00 – **Closing Ceremony & Reflection**:
    - Awards for categories e.g. "Best Use of Data", "Most Impactful Project", "Best Presentation"
    - Reflection circle: students share takeaways, surprises, and future plans
    - Certificates distributed and final group photo taken

Beyond PEAR: What Next?

- PEAR doesn't end on Demo Day. Students receive a Certificate of Completion and a template blurb to showcase their experience on CVs, LinkedIn or scholarship applications. We'll also follow up in the months after the programme to check in on students' growth.
- Participants will also receive a curated list of relevant opportunities, including:
    - Competitions and challenges
    - HALI Student Programs (High-Achieving, Low-Income)
    - Scholarship and college prep resources
- These materials are intended to help students translate their PEAR experience into long-term impact - whether through deeper research, real-world problem solving, or higher education pathways.

- Furthermore, students will have the opportunity to become **PEAR Ambassadors**, taking on a leadership role to represent the programme in their schools and communities, share their experiences, and help recruit and mentor future cohorts.

Contingency Planning

- As with any program, problems may occur. Here are some potential challenges as well as our mitigation strategies:
    - **Hardware fails**: We will have some **backup** kits, plus a troubleshooting guide. Mentors will be available to diagnose the issue and resolve it.
    - **Wi-Fi fails**: Install **Visual Studio Code** on all PCs as a Backup, we will also download Colab notebooks in advance and have printouts of all the build guides and reference sheets.
    - **Student is absent**: We will assign a **buddy** system for daily catch-ups. Each person will be responsible for their buddy's whereabouts during the day.
    - **Student is late**: The program is structured in such a way that missing the first 1-2 hours of a day will not be detrimental to their progress. The hands-on workshops will typically start from around 11.00 AM, rather than 9:00 AM. In addition, there are office hours later in the day, where each student can receive personalized support if they missed any program component.

Learning Tools and Digital Infrastructure

To streamline learning and ensure accessibility, all digital work is supported through lightweight and familiar tools:

1. Google Drive (Team Folders)

- Each team is assigned a shared folder for storing and organising:
    - Starter code and build templates
    - Circuit diagrams
    - Sensor data logs
    - Poster drafts and one-pagers
- Facilitators and mentors have edit/comment access for ongoing feedback

2. Google Colab

- All coding activities are completed in Colab notebooks
- No setup or installation is required, ensuring smooth participation across all devices
- Mentors can leave comments directly in cells, and students can collaborate in real time

3. Circuit Diagrams and Build Materials

- All builds are supported by visual **Fritzing diagrams (PDFs)**
- These diagrams are printed and provided during each build, and also available digitally in Drive folders
- Kits are pre assembled and verified before programme launch to avoid hardware issues during sessions

4. Google Slides

- Used for team collaboration on capstone posters and demo day presentations
- Mentors provide feedback during design stages to improve clarity and impact
- Visuals and structure are refined iteratively in preparation for final presentations

5. Google Forms

- Used to collect peer and mentor feedback during capstone development
- Forms include rubrics for clarity, creativity, technical execution, and communication
- Feedback is summarised and shared with each team for review and iteration

Mentorship and Facilitation Support

- Led by two Lead Facilitators (Tony and Emmanuela) and supported by three dedicated mentors
- **Daily Office Hours (16:00–17:00)** are available for one-on-one troubleshooting, feedback, or enrichment discussions
- Students are encouraged to seek help proactively and document blockers in their capstone journals for review
- Mentors will also receive a 1-page guidance sheet covering:
    - How to support **without solving** for students
    - How to give **constructive**, **kind** feedback
    - What their role is during **capstone development** and **Demo Day**

Media and Infrastructure

Media

- With **consent**, we'll document the programme via photos, short clips, student interviews, and a 2-minute recap video. This media supports future outreach, grant proposals, and stakeholder communication.
- We will rely on various social media platforms for publicity including **Instagram** and **LinkedIn**.

Required Hardware and Kits

- **10× SunFounder Raspberry Pi Pico 2W Kits** (1 per team of 3 students) plus extras
- USB-A/USB-C to Micro-USB cables
- Adapters
- Multimeters
- Extra Picos and Breadboards
- Printed circuit diagrams and build instructions for each group

Student Computing and Connectivity

- **30× PCs** (1 per student), Wi-Fi or Ethernet
- Google account access enabled for Colab, Drive, and Forms
- Browser-based work (no installations required).
- **Thonny IDE** and **VS Code IDE** will be Pre-installed.