Delivering a computing curriculum through the Coding Academy

Divya Joseph, Rashmita Panigrahi, Mamta Manaktala, Fiona Coventry, Claire Johnson, Robert Whyte
Raspberry Pi Foundation, Cambridge, UK

Short Abstract
The Coding Academy is a collaborative partnership between the Raspberry Pi Foundation and the Telangana Social Welfare Residential Educational Institutions Society (TSWREIS) to implement a comprehensive computing curriculum in Telangana. We report on approximately 460 students (grades 6 to 12) taking part in a curriculum at a government school that comprises computer science, information technology, and digital literacy. Student assessment data suggested that most students achieved pass grades, and educator feedback to date has been encouraging. However, observational data highlights the need for further content refinement, including simplifying language, adding more advanced activities for older students, and localising more examples.

Section 1 – Background

The growth of programming and digital literacy is a significant national priority in India since the publication of the National Education Policy of India 2020 (Ministry of Human Resource Development, 2020). Since 2023, the Raspberry Pi Foundation (RPF) has collaborated with the Telangana Social Welfare Residential Educational Institutions Society (TSWREIS) to introduce a computing curriculum spanning grades 6 to 12 at the Coding Academy in Moinabad, Telangana (Manaktala, 2023). Established by TSWREIS, the Coding Academy aims to provide high-quality education to students from scheduled castes and tribes in Telangana. The partnership aims to democratise computing education and foster digital empowerment among young people, particularly those from historically marginalised backgrounds.

Section 2 – Implementation

Content and pedagogical aspects

Like the CSpathshala curriculum (Shah, 2019), our computing curriculum includes a progression of interconnected computing topics, including programming, data science, information technology, and digital literacy (see Figure 1). Delivered by five trained educators with expertise in computer science, the curriculum entails 4.5 to 6 hours of computing sessions per week for students in each grade, equivalent to 80 to 130 sessions throughout the academic year.

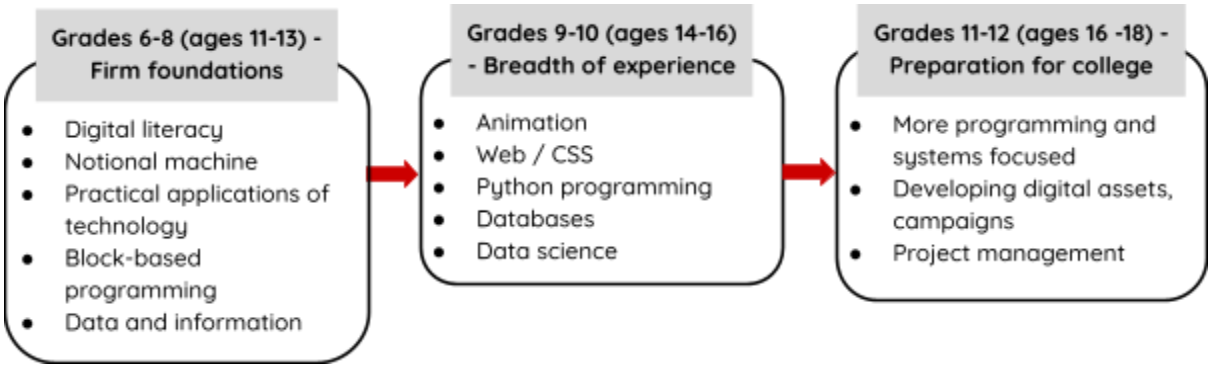


Figure 1: Curriculum progression

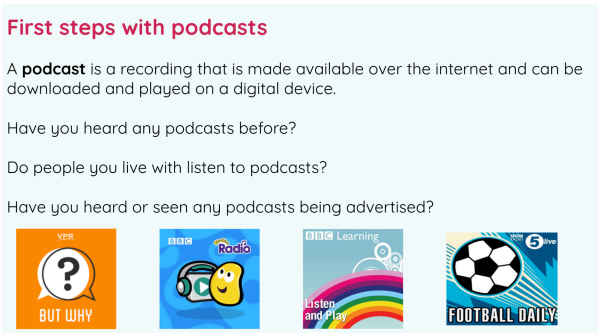
The curriculum follows a spiral design (Bruner, 1960) enabling learners to encounter concepts multiple times. Teaching materials include lesson plans, slides, and assessments. With each iteration, the complexity of the concept increases and a new context is introduced in which learners can apply their prior learning.

We adopted a pedagogical approach (Raspberry Pi Foundation, 2021, p.5) of 12 key principles underpinned by research in computing education (e.g. Sentance & Waite, 2021) where each principle has been shown to contribute to effective teaching and learning. Our educators were given initial training on lesson delivery and the pedagogical principles to deliver the curriculum.

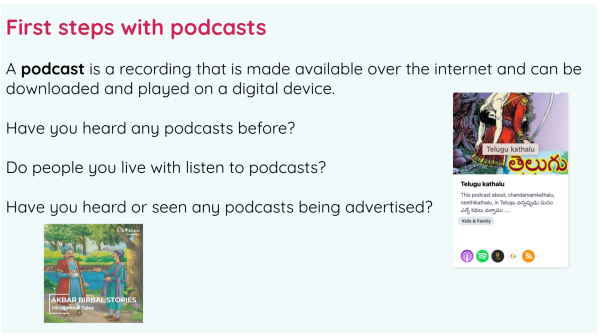
Figure 2: 12 pedagogical principles (Raspberry Pi Foundation, 2021)

Context

We assessed students' baseline interests and prior learning experience. Using these insights, our curriculum team adapted the content by localising examples which are region specific and relatable for the learners. Examples include (i) presenting students with examples of Indian podcasts, like "Stories of Akbar and Birbal" and podcasts in Telugu suited to the regional language of learners (see Figure 3) and (ii) adapting the data values of a spreadsheet activity to include cake recipes of local cuisines of Telangana (see Figure 4). We observed curriculum delivery and collected student responses to further refine the content.

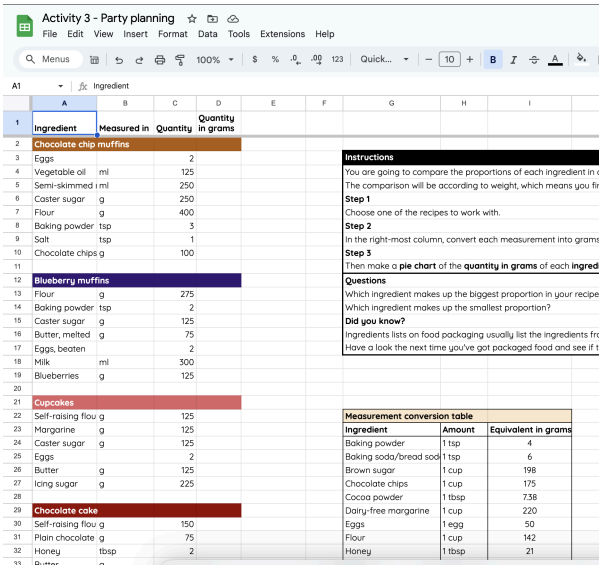


Original content



Adapted content

Figure 3: Creating media - audio production (Grade 7)



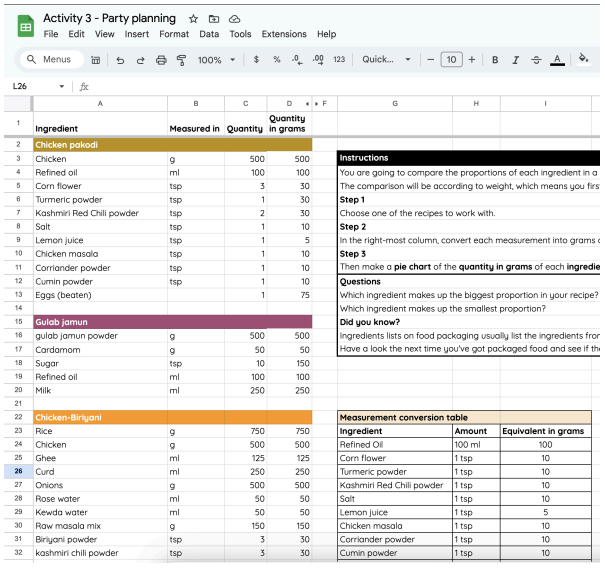


Figure 4: Modelling data using spreadsheet (Grade 9)

Section 3 – Impact

The curriculum's effectiveness and impact on student learning were regularly measured through formative and summative assessments. We collected quantitative data (assessment marks) and qualitative data (educator interviews, lesson observations) to understand the impact of the project. On student assessment, we found that 92% of student assessment marks (*n*=1939) represented at least a pass grade (40% of the available marks) while 72% of student assessment marks (*n*=1520) represented a higher grade of at least 60% of available marks.

Based on qualitative educator feedback data, there are several areas where further localisation of practice would be beneficial. Educators felt that the current content was more suitable for the lower grades and that the content and activities (e.g. Python programming) needed to be more advanced for the higher grades and matched to students' prior knowledge. Although educators felt confident in delivering the material, most requested additional training in how to apply the pedagogical principles (see Figure 2) and safeguarding in the classroom.

Other necessary adaptations referred to included ensuring that English text in the content and assessments is simplified (or translated) as some pupils found the language difficult to understand; educators also recommended changing the format of assessments for younger students to more accessible multiple choice style questions, and adding practical assessments.

Educators expressed a need to further localise content and examples used in the teaching material so that it was more culturally relevant. One facilitator supplemented the examples from the lesson plan with some in their local language. All educators agreed that they felt confident to adapt lesson plans if needed based on students' responses or because lesson content could not be covered in the allocated time.

Challenges

The Coding Academy grappled with challenges ranging from limited computer and internet access, and unreliable electricity disrupting learning sessions. Future adaptations to address these challenges with equipment and infrastructure include the provision of alternative, shortened activities for topics and more printed materials, unplugged activities and exercise books. Diverse student backgrounds, spanning from novice to experienced, also demanded tailored approaches to accommodate varying levels of prior exposure to technology. Crafting inclusive lesson plans that effectively engage all students proved to be a significant challenge, given the spectrum of student experiences—ranging from familiarity with different frameworks to new admissions lacking prior computing background.

Section 4 – Conclusion

Results from the first year implementation have been promising with educators commenting on the quality of the training and curriculum ("[It was] more than what [they were] expecting") and many students indicating that they wanted to continue studying computing in the future. We did note some areas for further work. In particular, we intend to make further adaptations to both content, pedagogy and training to address challenges educators and students face, such as limited technology access, language barriers, and varying prior experience.

Section 5 Bibliography

Bruner, J. S. (1960). *The process of education*. Harvard University Press.

Manaktala, M. (2024, October 31). *Coding futures: Celebrating our educational partnership in Telangana*. Raspberry Pi Foundation. https://rpf.io/blog-tcace

MHRD (2020, August 5). National Education Policy 2020. Retrieved from https://www.education.gov.in/sites/upload_files/mhrd/files/NEP_Final_English_0.pdf

Raspberry Pi Foundation (2021). Big Book of Programming Pedagogy. https://rpf.io/ped-book

Sentance, S., & Waite, J. (2021). *Teaching programming in schools: A review of approaches and strategies*. Raspberry Pi Foundation. https://rpf.io/teaching-programming-in-schools

Shah, V. (2019). CSpathshala: bringing computational thinking to schools. *Communications of the ACM*, *62*(11), 54–55. https://doi.org/10.1145/3343445