



Year 8 – Mobile app development

Unit introduction

In a world where there's an app for every possible need, this unit aims to take the learners from designer to project manager to developer in order to create their own mobile app. Using App Lab from code.org, learners will familiarise themselves with the coding environment and have an opportunity to build on the programming concepts they used in previous units before undertaking their project. Learners will work in pairs to consider the needs of the user; decompose the project into smaller, more manageable parts; use the pair programming approach to develop their app together; and finish off by evaluating the success of the project against the needs of the user.

Overview of lessons

Overview of lessons

| Lesson | Brief overview | Learning objectives |
|-----------------|--|--|
| 1. App for that | This lesson will introduce learners to the unit. They will get their creative minds going by thinking of an app that could do good in the world. They will then consider how decomposition is an important process to follow when tackling a large problem such as creating an app. Learners will be introduced to the app they will be developing over the coming lessons and given an opportunity to | <ul style="list-style-type: none">• Identify when a problem needs to be broken down• Implement and customise GUI elements to meet the needs of the user |

| | | |
|-----------------------|--|--|
| | decompose the problem as well as to sign into code.org to become used to the App Lab environment. | |
| 2. Tappy Tap App | Learners will be introduced to the concept of event-driven programming and applying the paradigm to the app they started to develop last week. They will be shown the coding environment and the first steps will be taken using live coding, in which the learners will write their code alongside the teacher. Learners will then use the pedagogical approach of pair programming to work through a set of instructions and challenges to develop the app further. | <ul style="list-style-type: none"> ● Recognise that events can control the flow of a program ● Use user input in an event-driven programming environment ● Use variables in an event-driven programming environment ● Develop a partially complete application to include additional functionality |
| 3. School Lab Studios | In this lesson, learners will be presented with an app that has three errors. They will have to open the app to attempt to spot and fix the errors. Next, the learners will work on the score screen of the Tappy Tap App, to make it display the user's score at the end of the game. Once this app is complete the class will be presented with the project that they will work on for the remaining lessons of the unit. In their pairs, learners will choose a project to complete and agree upon success criteria before being given screen designs to complete for homework. | <ul style="list-style-type: none"> ● Identify and fix common coding errors ● Pass the value of a variable into an object ● Establish user needs when completing a creative project |
| 4. User input | In this lesson, the learners will start by thinking about how user input is captured and processed, before being given the challenge of adding code to a prebuilt app to deal with user input. Learners will then decompose the app project that they started last lesson into more manageable steps. Using the pair programming approach, learners will then start to develop their app by working through their decomposed steps. At the end of the lesson, learners will document and reflect on their progress and make a plan for the following lesson. | <ul style="list-style-type: none"> ● Apply decomposition to break down a large problem into more manageable steps ● Use user input in a block-based programming language ● Use a block-based programming language to create a sequence |

| | | |
|-----------------------|--|--|
| | | <ul style="list-style-type: none"> • Use variables in a block-based programming language |
| 5. App development | <p>The main focus of this lesson is to spend most of the time developing the learners' app projects further. The learners will start by recapping their work and what they planned in the previous lesson. They will then spend time building their apps using pair programming. Towards the end of the lesson, the learners will ask classmates to review their apps in order to get feedback that they can respond to in the next lesson.</p> | <ul style="list-style-type: none"> • Use a block-based programming language to include sequencing and selection • Use user input in a block-based programming language • Use variables in a block-based programming language • Reflect and react to user feedback |
| 6. Project completion | <p>This is the final lesson of the unit and the focus will be on completing and evaluating the learners' app projects. The lesson starts with an activity to remind the learners about problem-solving and debugging, followed by a short activity to help them plan the time that they have left in the lesson to complete their app. The learners will be given time to complete their apps before evaluating their success against their criteria and by answering a short set of questions. The last 15 minutes of the lesson will be used for the learners to take an individual assessment through a multiple-choice test.</p> | <ul style="list-style-type: none"> • Use a block-based programming language to include sequencing and selection • Use user input in a block-based programming language • Use variables in a block-based programming language • Evaluate the success of the programming project |

Progression

This unit progresses students' knowledge and understanding of programming constructs in a block-based programming environment. Learners will also develop their computational thinking and project planning, by going from decomposing a larger project into smaller parts and creating success criteria for the project to getting user feedback and evaluating their projects.

Please see the learning graph for this unit for more information about progression.

Curriculum links

[National curriculum links](#)

- Design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables, or arrays]; design and develop modular programs that use procedures or functions
- Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem
- Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability

Assessment

Summative assessment

- Please see the assessment question and answer documents for this unit.

Assessment rubric

- Please see the assessment rubric document for this unit.

Subject knowledge

This unit focuses on the development of the following key techniques:

- Event handling
- Sequencing
- Variables
- Selection
- Operators

App Lab by code.org is used throughout the unit, so it is important that you are comfortable with the language and environment. In order to get a feel for the level of skill required, why not try the activities in the unit yourself before using them with your learners?

For this unit, it is essential that the teacher has registered for a teacher account at code.org and is able to set up a class ahead of any lessons taking place. Setting up a class allows the teacher to create login details for each learner, which can be printed and distributed in the lesson. The class manager will also allow the teacher to track the progress of the learners through the unit. Guidance for how to set up a class can be found here:

- [App Lab classroom instructions](https://ncce.io/prg3-1-rc) (ncce.io/prg3-1-rc)

Enhance your subject knowledge to teach this unit through the following training opportunities:

Online training courses

- [Programming Pedagogy in Secondary Schools: Inspiring Computing Teaching](https://ncce.io/secondarypedagogy) (ncce.io/secondarypedagogy)
- [Programming 101](https://ncce.io/prog101) (ncce.io/prog101)
- [Programming 102](https://ncce.io/progr102) (ncce.io/progr102)

Face-to-face courses

- [Key Stage 3 Computing for Non-Specialists](https://ncce.io/f2fks3) (ncce.io/f2fks3)

Resources are updated regularly — the latest version is available at: ncce.io/tcc.

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see ncce.io/ogl.