

[CS in Schools](#) - Outline of Year 7 Coding Syllabus

Overall Student Learning Objectives:

- Students gain an introductory understanding of programming using a text-based coding language
- Students experience success in coding small programs and completing exercises
- Students incorporate visual and literacy based creative elements into their programs
- Students have fun and gain confidence with programming and appreciate the relevance it has in their lives

The table below outlines the 8 lesson sequence that comprise this unit of learning. The headings also links to the respective lesson folders.

If you “run out of lessons” or should the volunteer and/or teacher be absent, please refer to the [Supplementary Lessons guide](#) below for additional lessons and resources.

Lesson	Learning Objectives
1	Introduction to CS in Schools <ul style="list-style-type: none">• Introducing your Industry Volunteer• Be signed up and enrolled in Repl.it course• Writing and understanding your first Program: Hello, world!• Playing and modifying an existing program
2	Displaying Text on the Screen and Input <ul style="list-style-type: none">• Introduction to “Whitespace” in code• Understand what Error Messages are and how they help• Learn how programs flow• Print() - Displaying text on the screen• Input() - Used to pause and wait for “Enter” key
3	Colour your world! <ul style="list-style-type: none">• Display text in different colours, highlights and styles• Use and understand how the “+” symbol concatenates style constants to strings
4	Input and Introduction to Variables <ul style="list-style-type: none">• Be introduced to Variables• Accept string Input() from a user and store it into a variable. Use of the “=” character to assign a value to a variable• Using the print() with “+” character to display the value inside the variable on the screen

Lesson	Learning Objectives
5	<p data-bbox="243 180 674 207"><u>Programs that Make Decisions</u></p> <ul data-bbox="296 217 1392 354" style="list-style-type: none"><li data-bbox="296 217 1392 245">● Introduction to Flowcharts - How they are used to represent the flow of a program<li data-bbox="296 253 1392 280">● IF statements - How we can use them to make choices in a program<li data-bbox="296 289 1392 316">● Introduction of “==”, used to compare if one expression is equivalent to another.<li data-bbox="296 324 1392 354">● The importance of indentation in IF statements
6	<p data-bbox="243 391 684 418"><u>Round and round we go (loops)</u></p> <ul data-bbox="296 428 1094 526" style="list-style-type: none"><li data-bbox="296 428 1094 456">● How loops are represented in flowcharts<li data-bbox="296 464 1094 492">● Using GOTO as an introduction to how loops work in code<li data-bbox="296 500 1094 526">● Understanding != comparison
7	<p data-bbox="243 566 632 594"><u>Introduction to Assignment</u></p> <ul data-bbox="296 604 1566 667" style="list-style-type: none"><li data-bbox="296 604 1566 631">● Be provided with an outline of the assignment rubric, code examples, template and video guide<li data-bbox="296 639 1566 667">● Commence working on the assignment
8	<p data-bbox="243 703 632 730"><u>Working Lesson and Finale</u></p> <ul data-bbox="296 740 852 837" style="list-style-type: none"><li data-bbox="296 740 852 768">● Continuing working on the assignment<li data-bbox="296 776 852 803">● Next steps beyond CSinSchools Year 7<li data-bbox="296 812 852 837">● Farewells!

CS in Schools - Overview of Lesson Format

Each of the lessons in the core syllabus generally follow the same format.

They each address the 5 Es of Learning: Engage, Explore, Explain, Elaborate, Evaluate

Section	Description
Learning Objectives	An overview of key learning goals is outlined
Engage - Coding Demo	Pre-written code, related to this lesson's topic, is demoed to the students
Explore - Student Tweaking of Code Demo	Students tweak the demoed code to customise it.
Explain - Theory and Concepts	Formal concepts and explanations are taught. Co-construction of programs are done here.
Elaborate - Exercise 1	Students undertake a series of 3 exercises that allows them to gradually construct and demonstrate their understanding of the content: <ul style="list-style-type: none">• Exercise 1 is a scaffolded activity that requires "fill-in-the-blanks" type coding to provide a gentle introduction to individual construction.• Exercise 2 requires students to complete some pre-written code.• Exercise 3 requires students to write code from scratch.
Elaborate - Exercise 2	
Elaborate - Exercise 3	
Explain and Elaborate (as needed) - Walkthrough video of Exercise 3	Video walkthroughs of solutions are provided for Exercise 3. This is useful for students who wish to revisit topics to clarify their understanding. It can be also be useful to catch students up who may have missed classes etc.
Explore and Extended - Extension Exercise	Students who complete all the exercises quickly and easily and have demonstrated mastery of the concepts covered, can undertake the optional extension exercises to deepen and/or broaden their understanding.
Evaluate - Show and Tell, Formal Reflection	Students present and talk about their work and learning process to the class, linking it back to the topic(s) covered.

Guide to Supplementary Lessons

The optional six supplementary lessons outlined below are designed to augment the core lessons in the syllabus. They can be used, for example:

- When you “run out of lessons” from the core syllabus:
 - Any of the supplementary lessons can be taught
- When the industry volunteer is absent:
 - Any of the lessons can be used, though some require regular teacher facilitation
- When only a substitute teacher is present:
 - Provided prerequisites have been met, students can instructed to complete the self-paced lessons

Lesson	Topic	Prerequisites	Regular Teacher Facilitation	Involves Coding	Self Paced
1	<u>Artificial Intelligence Discussion</u> <ul style="list-style-type: none"> • Watch videos and engage in a classroom discussion following pre-defined rules. 	None	✓		
2	<u>Unplugged Activities</u> <ul style="list-style-type: none"> • Involves moving about and “acting” out various coding concepts 	None	✓		
3	<u>Exploring, Analysing and Modifying Code</u> <ul style="list-style-type: none"> • Students load up and tweak pre-written Repl projects 	None, though those who have completed Lessons 5 & 6 will be able to write code		✓	
4	<u>Animating Text</u> <ul style="list-style-type: none"> • Students watch a walkthrough video and follow along 	Completed Lessons 1 & 2. <i>There will be code covered in the video that students may not yet understand, but they can just follow along.</i>		✓	✓
5	<u>Multiple Choice Quiz</u> <ul style="list-style-type: none"> • A 25-question multiple choice quiz to be completed individually. 	Completed Lessons 1 - 6			✓

6	<p><u>Program Flow</u></p> <ul style="list-style-type: none">• Learning more about program flow including ELSE and !=	Completed Lesson 5			
---	---	--------------------	---	---	--