CODE AVENGERS

AUSTRALIAN FEDERAL CURRICULUM (V9)

DIGITAL TECHNOLOGIES ALIGNMENT

AUSTRALIAN FEDERAL CURRICULUM (V9)
DIGITAL TECHNOLOGIES ALIGNMENT

The Australian Federal Curriculum Version 9 for Digital Technologies covers learning from Foundation to Year 10. Various Australian states have their own intermediary curriculum documents, particularly Victoria, New South Wales, and Western Australia. These states repackage the Australian Curriculum, and may have more or less content in some learning areas. However, for the most part, the same content is covered, so this alignment document is a fair representation of alignment for all states from Foundation to Year 10. Each state also has its own senior curriculum and qualifications framework for years 11-12, so the Federal Curriculum only reaches up to Year 10. Our existing Australian senior secondary alignments can be found here.

This report summarizes the alignment of the Code Avengers platform resources with these Federal Curriculum standards. The curriculum content comprises 2 strands, with varying substrands, that span across 6 age bands from F-10. **Note:** not every grade has content for all substrands, so this report will group content into the two strands.

Strands & Substrands

1. **Knowledge & Understanding**
   a. Digital Systems
   b. Data Representation

2. **Processes & Production Skills**
   a. Acquiring, managing and analysing data
   b. Investigating and defining
   c. Generating and designing
   d. Producing and implementing
   e. Evaluating
   f. Collaborating and managing
   g. Privacy and security

Grade bands:

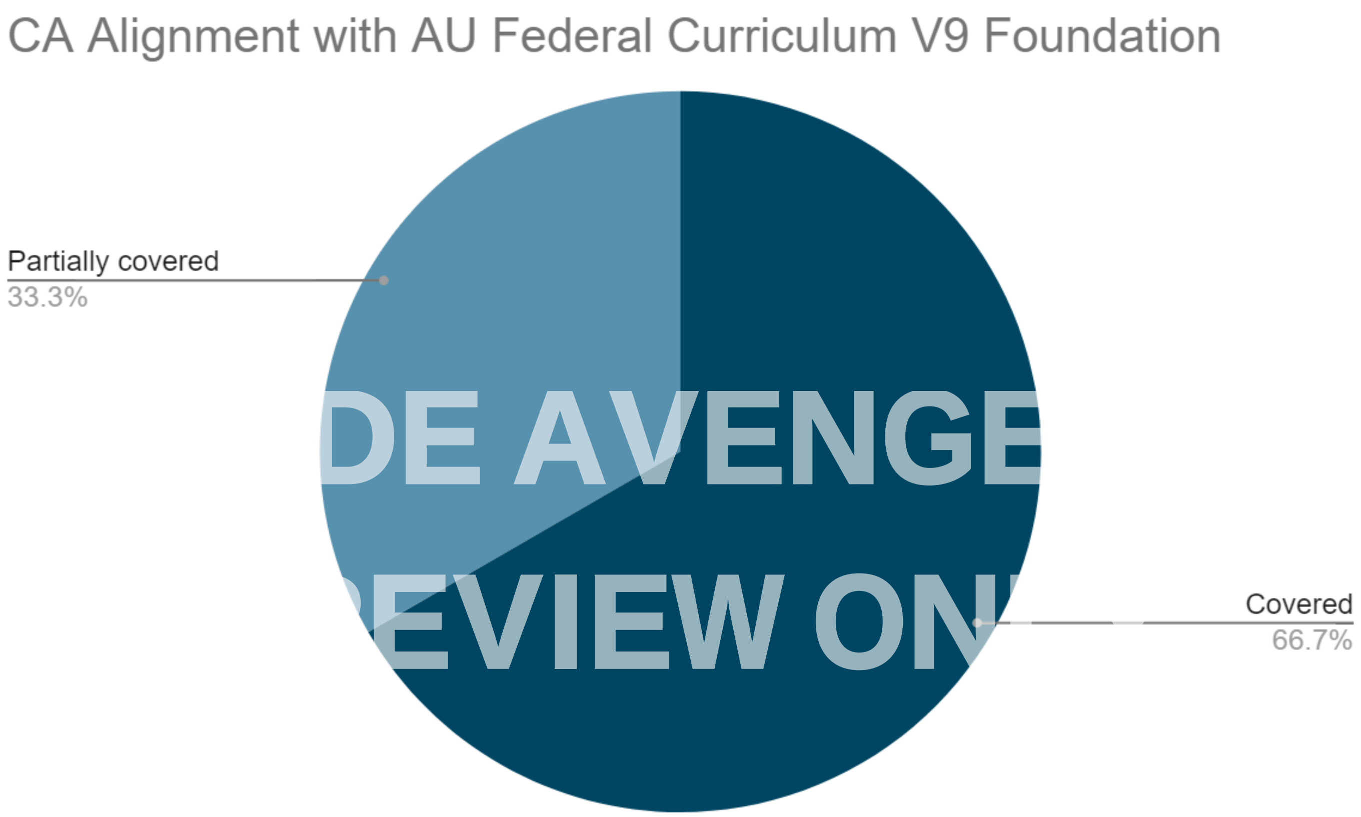
1. Foundation
2. Years 1-2
3. Years 3-4
4. Years 5-6
5. Years 7-8
6. Years 9-10

There are 73 content descriptors/curriculum standards from Foundation to Year 10. The tables and charts on the following pages will highlight how well the Code Avengers content currently available aligns with these curriculum standards.

Alignment by Grade Band

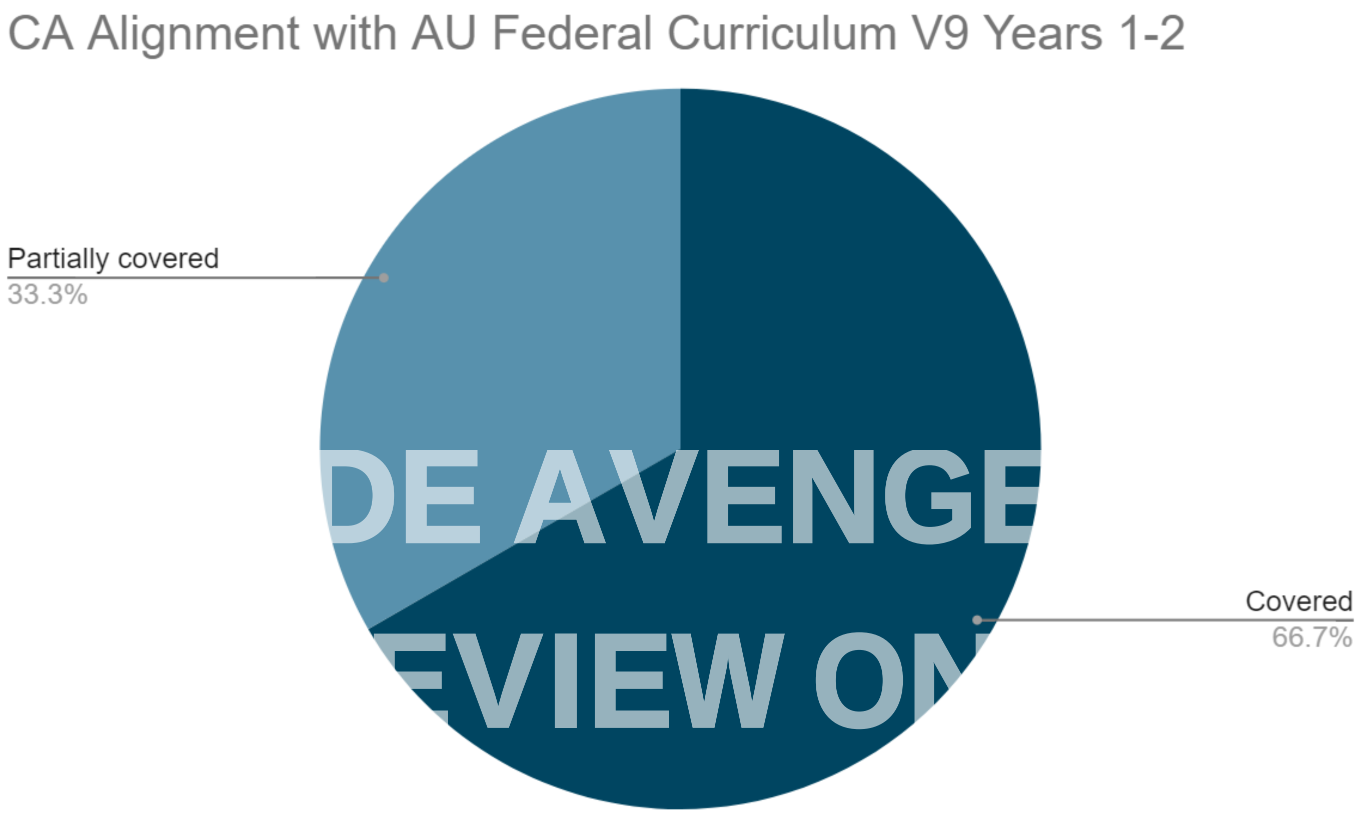
Foundation

Topic	Yes	Partial	No
1. Knowledge & Understanding	2	0	0
2. Processes & Production Skills	0	1	0
Totals (out of 3 standards)	**2**	**1**	**0**



Years 1-2

Topic	Yes	Partial	No
1. Knowledge & Understanding	2	0	0
2. Processes & Production Skills	4	3	0
Totals (out of 9 standards)	**6**	**3**	**0**



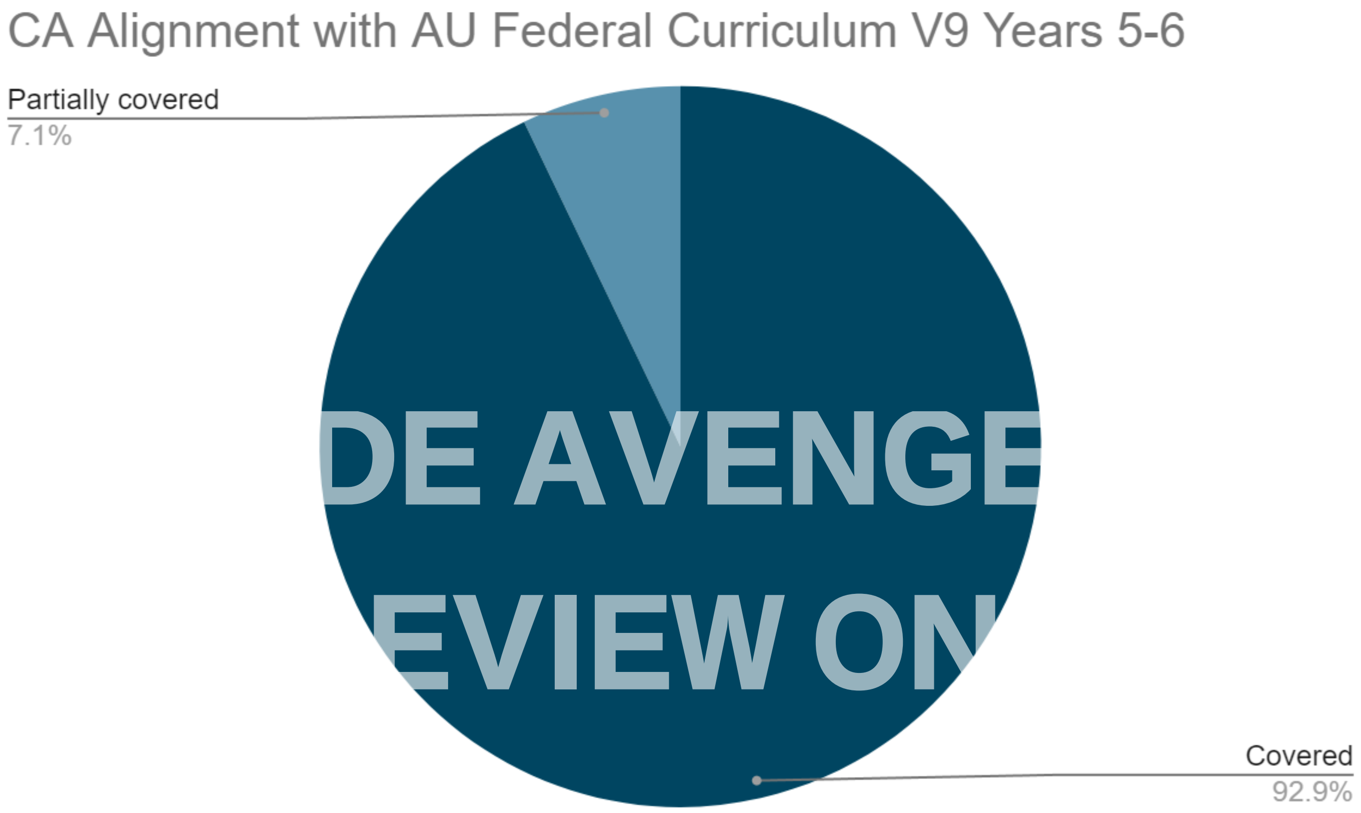
Years 3-4

Topic	Yes	Partial	No
1. Knowledge & Understanding	3	0	0
2. Processes & Production Skills	9	0	0
Totals (out of 12 standards)	**12**	**0**	**0**

CA Alignment with AU Federal Curriculum V9 Years 3-4

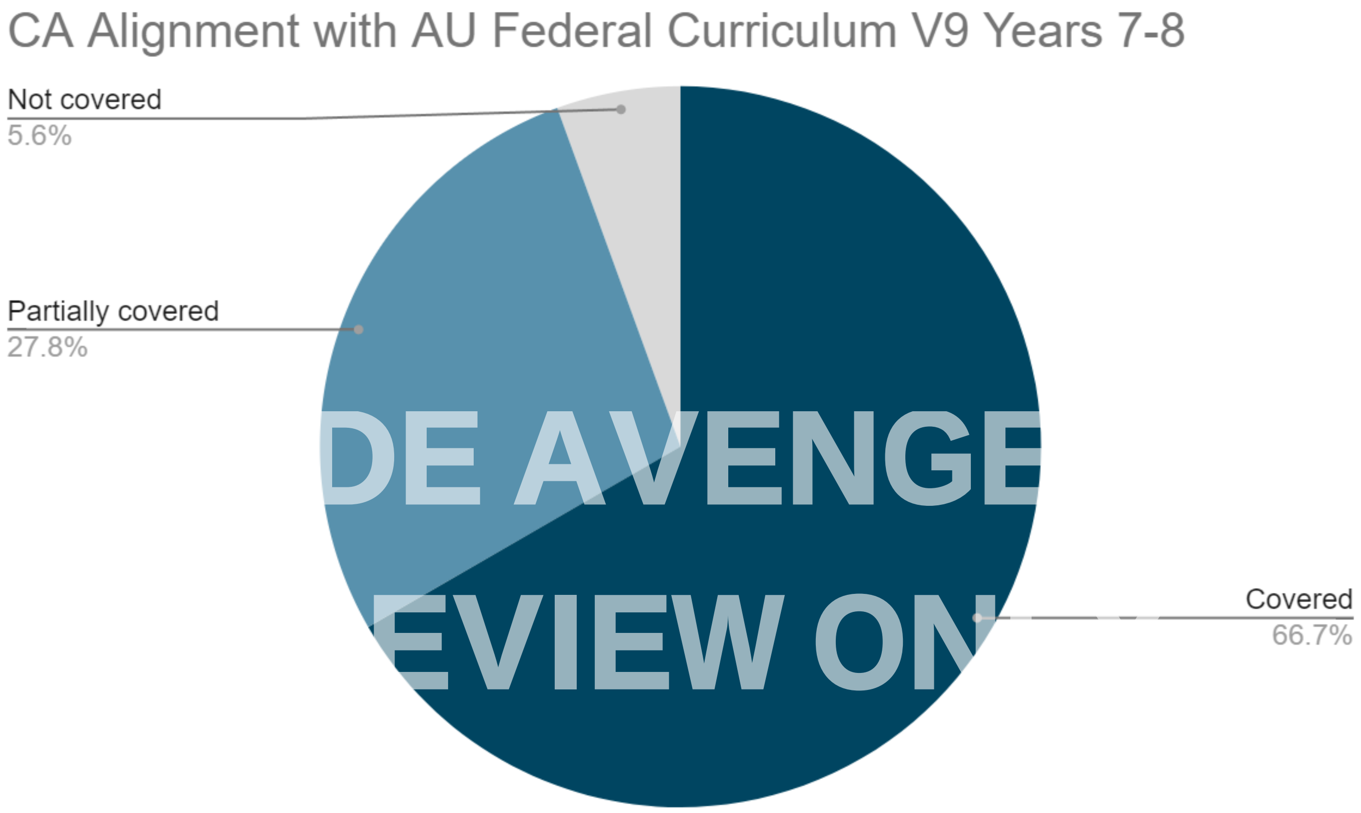
Years 5-6

Topic	Yes	Partial	No
1. Knowledge & Understanding	3	1	0
2. Processes & Production Skills	10	0	0
Totals (out of 14 standards)	**13**	**1**	**0**



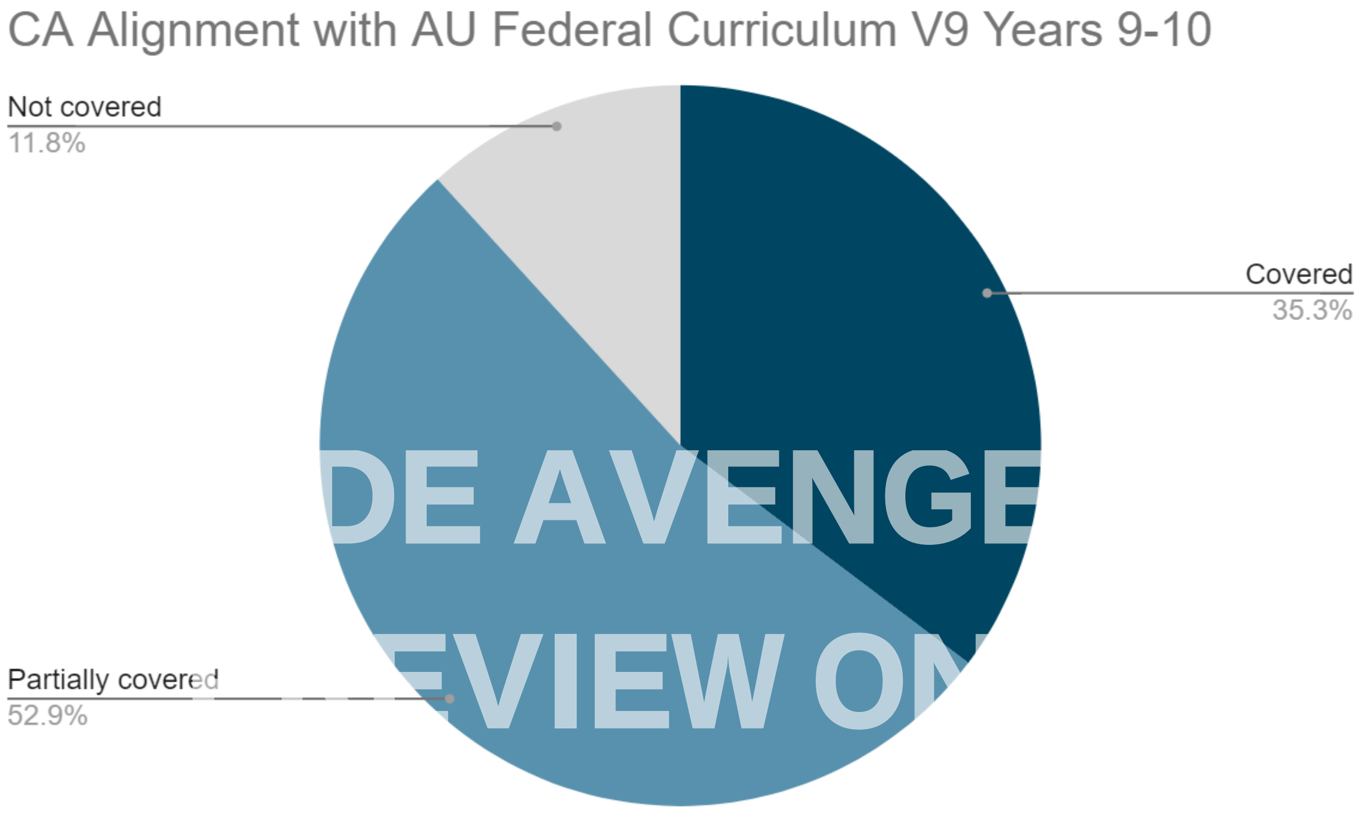
Years 7-8

Topic	Yes	Partial	No
1. Knowledge & Understanding	3	0	1
2. Processes & Production Skills	9	5	0
Totals (out of 18 standards)	**12**	**5**	**1**



Years 9-10

Topic	Yes	Partial	No
1. Knowledge & Understanding	2	0	1
2. Processes & Production Skills	4	9	1
Totals (out of 17 content descriptors)	**6**	**9**	**2**



Appendices follow

Appendix 1: Background & Detail

Code Avengers content covers the Australian Federal Curriculum V9 for Digital Technologies standards as follows: out of a total of 73 standards, we fully cover 51, partially cover 19, and don't cover 3.

- We fully cover the Years 3-4 band.
- We fully or partially cover all standards F-6.
- At Years 7-8, we cover or partially cover 94.4% of standards.
- At Years 9-10, we cover or partially cover 88.2% of standards.

A full breakdown of the alignment to the standards is included below.

Key

Color	Explanation
🟩	Code Avengers content covers this standard
*	Code Avengers partially covers this content, and/or it is content that would be covered organically through participation in Digital Technologies learning e.g. logging into school accounts, using devices, and so on.
🟨	Code Avengers content partially covers this standard.
🟥	Code Avengers content does not cover any of this standard at present.

AU Federal (V9) DT Alignments		
1.	**Foundation**	
1.1.	**Knowledge and understanding**	
1.1.1.	Recognise and explore digital systems (hardware and software) for a purpose	🟩
1.1.2.	Represent data as objects, pictures and symbols	🟩
1.2.	**Processes and production skills**	
1.2.1.	Identify some data that is personal and owned by them	🟨
2.	**Years 1-2**	
2.1.	**Knowledge and understanding**	
2.1.1.	Identify and explore digital systems and their components for a purpose	* 🟩
2.1.2.	Represent data as pictures, symbols, numbers and words	🟩

2.2.	**Processes and production skills**	
2.2.1.	Investigate simple problems for known users that can be solved with digital systems	🟩
2.2.2.	Follow and describe algorithms involving a sequence of steps, branching (decisions) and iteration (repetition)	🟩
2.2.3.	Discuss how existing digital systems satisfy identified needs for known users	🟩
2.2.4.	Use the basic features of common digital tools to create, locate and communicate content	🟨
2.2.5.	Use the basic features of common digital tools to share content and collaborate demonstrating agreed behaviours, guided by trusted adults	🟨
2.2.6.	Access their school account with a recorded username and password	*
2.2.7.	Discuss that some websites and apps store their personal data online	🟨
3.	**Years 3-4**	
3.1.	**Knowledge and understanding**	
3.1.1.	Explore and describe a range of digital systems and their peripherals for a variety of purposes	🟩
3.1.2.	Explore transmitting different types of data between digital systems	🟩
3.1.3.	Recognise different types of data and explore how the same data can be represented differently depending on the purpose	🟩
3.2.	**Processes and production skills**	
3.2.1.	Define problems with given design criteria and by co-creating user stories	🟩
3.2.2.	Follow and describe algorithms involving sequencing, comparison operators (branching) and iteration	🟩
3.2.3.	Generate, communicate and compare designs	🟩
3.2.4.	Implement simple algorithms as visual programs involving control structures and input	🟩
3.2.5.	Discuss how existing and student solutions satisfy the design criteria and user stories	🟩
3.2.6.	Use the core features of common digital tools to create, locate and communicate content, following agreed conventions	🟩

3.2.7.	Use the core features of common digital tools to share content, plan tasks, and collaborate, following agreed behaviours, supported by trusted adults	🟩
3.2.8.	Access their school account using a memorised password and explain why it should be easy to remember, but hard for others to guess	🟩
3.2.9.	Identify what personal data is stored and shared in their online accounts and discuss any associated risks	🟩
4.	**Year 5-6**	
4.1.	**Knowledge and understanding**	
4.1.1.	Investigate the main internal components of common digital systems and their function	🟨
4.1.2.	Examine how digital systems form networks to transmit data	🟩
4.1.3.	Explain how digital systems represent all data using numbers	🟩
4.1.4.	Explore how data can be represented by off and on states (zeros and ones in binary)	🟩
4.2.	**Processes and production skills**	
4.2.1.	Define problems with given or co-developed design criteria and by creating user stories	🟩
4.2.2.	Design algorithms involving multiple alternatives (branching) and iteration	🟩
4.2.3.	Design a user interface for a digital system	🟩
4.2.4.	Generate, modify, communicate and evaluate designs	🟩
4.2.5.	Implement algorithms as visual programs involving control structures, variables and input	🟩
4.2.6.	Evaluate existing and student solutions against the design criteria and user stories and their broader community impact	🟩
4.2.7.	Select and use appropriate digital tools effectively to create, locate and communicate content, applying common conventions	🟩
4.2.8.	Select and use appropriate digital tools effectively to share content online, plan tasks and collaborate on projects, demonstrating agreed behaviours	🟩
4.2.9.	Access multiple personal accounts using unique passphrases and explain the risks of password re-use	🟩

4.2.10.	Explain the creation and permanence of their digital footprint and consider privacy when collecting user data	🟩
5.	**Years 7-8**	
5.1.	**Knowledge and understanding**	
5.1.1.	Explain how hardware specifications affect performance and select appropriate hardware for particular tasks and workloads	🟥
5.1.2.	Investigate how data is transmitted and secured in wired and wireless networks including the internet	🟩
5.1.3.	Investigate how digital systems represent text, image and audio data using integers	🟩
5.1.4.	Explain how and why digital systems represent integers in binary	🟩
5.2.	**Processes and production skills**	
5.2.1.	Acquire, store and validate data from a range of sources using software, including spreadsheets and databases	🟩
5.2.2.	Analyse and visualise data using a range of software, including spreadsheets and databases, to draw conclusions and make predictions by identifying trends	🟩
5.2.3.	Model and query the attributes of objects and events using structured data	🟩
5.2.4.	Define and decompose real-world problems with design criteria and by creating user stories	🟨
5.2.5.	Design algorithms involving nested control structures and represent them using flowcharts and pseudocode	🟩
5.2.6.	Trace algorithms to predict output for a given input and to identify errors	🟩
5.2.7.	Design the user experience of a digital system	🟨
5.2.8.	Generate, modify, communicate and evaluate alternative designs	🟩
5.2.9.	Implement, modify and debug programs involving control structures and functions in a general-purpose programming language	🟩
5.2.10.	Evaluate existing and student solutions against the design criteria, user stories and possible future impact	🟨
5.2.11.	Select and use a range of digital tools efficiently, including unfamiliar features, to create, locate and communicate content, consistently applying common conventions	🟨

5.2.12.	Select and use a range of digital tools efficiently and responsibly to share content online, and plan and manage individual and collaborative agile projects	🟨
5.2.13.	Explain how multi-factor authentication protects an account when the password is compromised and identify phishing and other cyber security threats	🟩
5.2.14.	Investigate and manage the digital footprint existing systems and student solutions collect and assess if the data is essential to their purpose	🟩
6.	**Years 9-10**	
6.1.	**Knowledge and understanding**	
6.1.1.	Investigate how hardware and software manage, control and secure access to data in networked digital systems	🟥
6.1.2.	Represent documents online as content (text), structure (markup) and presentation (styling) and explain why such representations are important	🟩
6.1.3.	Investigate simple data compression techniques	🟩
6.2.	**Processes and production skills**	
6.2.1.	Develop techniques to acquire, store and validate data from a range of sources using software, including spreadsheets and databases	🟨
6.2.2.	Analyse and visualise data interactively using a range of software, including spreadsheets and databases, to draw conclusions and make predictions by identifying trends and outliers	🟨
6.2.3.	Model and query entities and their relationships using structured data	🟨
6.2.4.	Define and decompose real-world problems with design criteria and by interviewing stakeholders to create user stories	🟨
6.2.5.	Design algorithms involving logical operators and represent them as flowcharts and pseudocode	🟩
6.2.6.	Validate algorithms and programs by comparing their output against a range of test cases	🟩
6.2.7.	Design and prototype the user experience of a digital system	🟨
6.2.8.	Generate, modify, communicate and critically evaluate alternative designs	🟨
6.2.9.	Implement, modify and debug modular programs, applying selected algorithms and data structures, including in an object-oriented programming language	🟩

6.2.10.	Evaluate existing and student solutions against the design criteria, user stories, possible future impact and opportunities for enterprise	🟩
6.2.11.	Select and use emerging digital tools and advanced features to create and communicate interactive content for a diverse audience	🟨
6.2.12.	Use simple project management tools to plan and manage individual and collaborative agile projects, accounting for risks and responsibilities	🟨
6.2.13.	Develop cyber security threat models, and explore a software, user or software supply chain vulnerability	🟨
6.2.14.	Apply the Australian Privacy Principles to critique and manage the digital footprint that existing systems and student solutions collect	🟥

Appendix 2: Code Avengers Deep Dive

Why Code Avengers?

Background

Code Avengers is the commercial realization of the results of a prolonged and extensive Ph.D. research project that identified 29 core elements required to successfully teach coding. It is a comprehensive online education platform (https://codeavengers.com) that effectively caters to educational institutes and individual learners. Our platform provides interactive, gamified courses that teach computer programming, web development, computational thinking, digital skills, design, and more, through an engaging and effective, “learn-by-doing” approach.

Instruction & Support

Our language vocabulary is tailored to support ESOL learners, including strategies like having both written and spoken instructions and limiting non-jargon vocabulary to 2 thousand most common English words. As a web-based platform, screen reading software makes our courses accessible and all tasks work similarly and have a predictable layout, with audio available for most courses to help reduce literacy barriers.

Our Professional Development courses help teachers develop inclusive teaching practices and strategies to support the learning and achievement of diverse learners with specific needs using digital technologies. Through our support and contact functionality, educators have access to a team of PD facilitators and content developers who support them in a variety of ways, such as assisting them in curriculum development, using our platform to plan individualized instruction, and offering hybrid/ inclusive learning opportunities.

Student Engagement

Our course developers are qualified, experienced teachers, who are experts in both the content and pedagogy of Computer Science (CS). They have been instrumental in creating quality content that separates us from other platforms by effectively engaging learners from a diverse range of backgrounds. Our scaffolded learning methods build students’ skills gradually which means they are continuously challenged and engaged in the learning process.

At Code Avengers, our whole business is focused on providing innovative, future-focused CS education that empowers all learners to develop the digital skills they need to be the creators and innovators of the future. Our course developers have expertise in inclusiveness and diversity, and Code Avengers courses are therefore designed with culturally responsive learning in mind. Because our team is made up of a diverse range of people with different cultures, genders, and experiences, we are able to provide diverse learners with relatable learning contexts and meaningful activities.

We ensure underserved populations can see themselves reflected within the industry and learning environment. Our courses have a diverse cast of characters from varied cultures, countries, and families. At Code Avengers, we support and encourage people from all walks of life to get excited about being creators and innovators in the digital space. The majority of our content is available in English and Spanish, with a growing library of content in German, French, Japanese, Maori, Dutch, Portuguese, Russian, Chinese/Mandarin, and other languages planned as well. This provides additional support for English Language Learners. Code Avengers has been used by over 2 million students worldwide.

Promoting Learning

At its core, CS is about solving problems. Digital outcomes, products, and services are developed in order to find innovative solutions to problems and opportunities we face. That's why at Code Avengers, we integrate problem-solving skills into every learning resource.

Divergent thinking is supported in courses that require students to generate original ideas and examine a variety of viable solutions. Students can investigate tasks in a 'non-linear' manner using logic puzzles that explicitly teach decision-making and problem-solving processes on our platform. We aim to create an environment in which students can gain firsthand experience with the concepts of optimization, redundancy, and reliability, as well as their implications for technological system design, development, and maintenance.

Students are empowered to take an active role in choosing the courses that they complete. They have access to a whole course library, with filters that can help them select a pathway. They can complete a course from beginning to end, or jump to a specific lesson. The courses focus on students creating code to demonstrate their understanding of a concept. This means that they are not just watching a video or passively learning but actively engaging in the learning material to create and modify work. Our teacher training emphasizes strategies like "Pair Programming", "Try 3 Then Me", and the importance of choice and space to tinker. Our learning content complements these approaches.

Individualized Learning

Teachers love the self-paced Code Avengers curriculum as it encourages and enables students to be confident, independent learners. Our courses are perfect for students of any experience level and have many tools available for students with diverse learning needs. When designing courses, we use a variety of strategies to support students and ensure learning is accessible.

Students enjoy the Code Avengers platform because of all the features it offers including immediate feedback, helpful videos, pop-up vocabulary, hints, a built-in integrated development environment (IDE), achievement tracking, and other resources. All tasks are self-graded with real-time progress tracking provided on the teacher dashboard and class data can be exported to a spreadsheet at any time. Teachers can maintain control of class direction by assigning specific modules to class groups or individual students and simultaneously manage class differentiation needs.

Not only can our content be tailored to our students, but our teachers also benefit from the platform's adaptability. Teachers can learn, explore, and upskill alongside their students without having to be experts. Code Avengers makes it easy for teachers with any amount of experience to facilitate effective CS learning in their classrooms. It is a high priority of ours to provide teachers with a suite of resources to support the delivery and monitoring of student learning, including live student tracking, analytics, lesson plans, quizzes, offline activities, and more.

Collaboration

Unplugged activities aimed at helping students think critically and strategically are available to support many of the courses. Students take on specific roles typical in a development cycle to contribute to a project as a team and are taught to manage their time and resources to produce digital artifacts. There are multiple opportunities to collaborate on Code Avengers. Our Pro games and website are shareable and we have JS and HTML/CSS IDEs as practice spaces that allow sharing your code and the ability to download work.

We also provide support and guidance for teachers that includes effective pedagogy and teaching strategies to promote collaboration in Computer Science learning.

Collaboration is fundamental to our internal ethos. We encourage learners and teachers to provide feedback and make suggestions for how to improve our platform, and we actively seek both perspectives when identifying new resources for development.

Real-World Applications

Our courses offer authentic learning opportunities that propose real-world challenges. This allows students to apply their understanding of CS concepts through memorable and compelling applications. Students are expected to use higher-order cognitive processes to analyze problems using a range of skills, such as detail-oriented coding, teamwork, and big-picture, long-term thinking.

Our Junior courses are story-based, and the learning happens within relevant real-world scenarios. At the Pro (senior) level, students write code to build real-world programs and apps that have actual everyday applications.

The impacts of computing are an important topic addressed in many of our courses. Students design and iteratively develop computational artifacts for practical purposes, personal expression, and to address societal issues. They are required to critically evaluate computational artifacts and consider both their beneficial and harmful effects on society. At higher levels, they theorize about the future of the computing innovations that have altered our culture and discuss the legal and ethical implications of these innovations.

Career & Technical Education

Students transition to learning industry-standard programming languages, such as JavaScript and Python, and web languages such as HTML5 and CSS3, or build their theoretical knowledge with our Computer Science and Design courses. These are some of the most in-demand languages in the tech sector, whilst also being very beginner-friendly for both students and teachers. This means learners are given an excellent foundation to move into entry-level/internship-type roles, or on to further study.

Cross-Curricular Connections

Cross-curricular links help learners to make connections with a range of other domains including further STEM topics, social sciences, and the arts. As a company that is already well established in the field of EdTech, Code Avengers is especially well placed to continue to adapt our product to enhance cross-curricular learning. This positions us well to support schools and teachers with the introduction of CS learning as curricula begin to roll out across various states of the USA. We understand that teachers sometimes need to work CS into their classrooms in intelligent ways, so our courses are built to integrate into mainstream learning and connect concepts across the curriculum. When developing new content, cross-curricular connections are a fundamental part of the planning process. Some examples of cross-curricular links found in our existing content are:

- Design theory e.g. color theory.
- History e.g. the history of numbering systems, the evolution of agriculture with AI and agritech.
- Social sciences e.g. the ethics and social impacts of current and future technology.
- Language & literacy e.g. writing and literature-themed activities and writing prompts, procedural writing, constructing texts for specific audiences.
- Science e.g. the physics of light and sound.
- Math e.g. geometry, coordinates, statistics, financial literacy, and more.

Assessment

The Code Avengers learning platform has a number of assessment opportunities available for learners.

Automatic Checking and Validation
Every task on the platform is assessed through automatic checking and validation, to ensure the learner has completed the task properly. This means learners are unable to progress to the next step of their learning without demonstrating an understanding of the step they are currently on. Hints and solutions are available to support learners so that they can self-assess, and practice self-reflection to monitor their own learning. When students use these features, assessment data is generated and displayed as a color-coded progress report in the teacher dashboard view. This means that formative assessment happens naturally as a part of the learning process, and teachers can use this data to identify learners in need of further support without any additional assessment or evaluation work needed on their part.

Quizzes
Our platform also has built-in quizzes that support the recapping and consolidation of information learned in the courses. Lesson plans contain Kahoot quizzes, Google Forms, and other tasks that can also be used to assess student learning for formative or summative purposes.

Projects
Shorter projects that require similar skills are connected to many courses. These have less scaffolding in the instructions in order to support the application of skills learned in the courses. This allows students to try to solve problems more independently and provides teachers with some insight into how well learners have grasped core concepts.

Our step-by-step lesson plans are organized into units that culminate in a project where learners apply the skills they have gained. During this process, learners co-construct and frequently refer to a rubric based on success criteria that they have identified. This gives them the skills they need to be able to feed forward in both self and peer assessment.

Supporting materials
Additional resources such as unplugged activities, workshops, and more, are available to allow for a range of ways to develop and assess learners' understanding of the concepts in the courses.

Standard alignment

Assessment is embedded within all Code Avengers courses. This means that the same coverage, as shown in the curriculum alignment above, applies for both learning content and assessment. The additional resources such as quizzes, projects, and lesson plans that are wrapped around the course content offer opportunities for learning to be assessed in multiple ways.

Performance descriptors

All of our courses come with teaching guides that include detailed breakdowns of the curriculum links, learning outcomes, and task objectives for every lesson and task in the course. These teaching guides also include key vocabulary and definitions so that any misconceptions or misunderstandings can be addressed before teaching and learning are undertaken.

Our growing library of fully guided, step-by-step unit and lesson plans also includes clear curriculum links, learning outcomes, and project options for summatively assessing learning from the unit.

This makes it clear to teachers what learning is being assessed in each lesson, course, and task, and what learners would be expected to be able to do/perform upon completion of the lesson or course.

Assessment Equity

Throughout Code Avengers courses, real-time assessment is carried out using a wide variety of activity and widget types, with all concepts assessed more than once throughout a course, using different learning modes. Courses are designed with diverse learning styles in mind, which means learners get multiple opportunities to apply their learning, in a way that suits them.

Audio is provided throughout many of the courses, particularly those aimed at younger learners, to support those who may struggle with literacy demands, including English Language Learners. Videos are provided in many courses for older learners as well.

Our step-by-step lesson plans offer guidance for differentiated learning in order to support a diverse range of learners, from those who need additional support, to gifted learners. The self-paced nature of Code Avengers, and the ability to assign different courses and modules to various learners, further support teachers to deliver differentiated programs of learning and assessment.

Professional Development

No experience or specific professional development (PD) training is required in order to successfully use Code Avengers to support Computer Science learning in the classroom. However, we do offer a range of resources to support teachers to get the best out of using Code Avengers with their learners.

Our courses are scaffolded to support educators and learners of all experience levels. Teachers can complete courses themselves to upskill or learn alongside students collaboratively. The rich set of resources, with interactive tasks and activities, is ideal for teachers to learn computer science concepts, programming languages, and other applications of real-world skills.

In addition to the full course library, EDU licenses provide access to our interactive Professional Development courses, covering topics such as computer science concepts, collaborative teaching strategies and effective pedagogy for computer science, repositioning the teacher from being the expert to learning alongside their students, fostering perseverance and innovation, and culturally responsive pedagogy. This is valuable learning, given the significant number of teachers who come into teaching Computer Science without a background in the subject, and the importance of being able to engage diverse learners.

Code Avengers is an accredited PD provider, with extensive experience delivering in-person and online professional development programs in numerous settings across the globe. We at Code Avengers will always provide support on our platform to our customers, as an included part of our software as a service. We can also provide custom, personal, and expert professional development to any teacher, school, or district that could benefit from it, in-person or online.

We have consistently and effectively delivered high-quality PD since our inception. Teachers' confidence and time are a top priority, and we can provide additional tools beyond our included support to build confidence and save time.

PD Testimonial (Roz, Maihihi School):

> “This was an excellent piece of PD which was relevant and well facilitated. Thank you so much. The presenters were great. For myself, as a non-confident digital user, they were very considerate and understanding in helping me to work through what was to be taught and the best way to support my students with their digital learning. Thank you for giving me the confidence to be a digital learner and a digital teacher.“

Please contact the Code Avengers team to learn more about our current offerings and schedule. More testimonials are included below.

Research & Testimonials

Research & Case Studies

Queensland Pilot
In late 2019, a trial was conducted with Trinity Bay State High School in Manunda, Queensland Australia. It has a demographic of approximately 35% indigenous learners. Having previously had difficulty engaging their learners with digital technologies, the school piloted the Code Avengers platform to explore whether it would improve student engagement and learning outcomes.

The trial was a resounding success, with student engagement significantly improved, particularly for indigenous girls. The pilot report can be read here, summarized in the quote below with key findings and benefits of using Code Avengers:

> “Differentiation of Digital Technologies course content is now truly possible with the Code Avengers platform. We have a very large Special Needs Program at the school with over 150 students identified with special needs/cognitive disorders/physical issues. Many of these students have worked extremely effectively on the Code Avengers platform. This is an area where their various disabilities are of little consequence. Many of the specialist staff that support them have been extremely surprised at the results that the students have achieved using Code Avengers.”

Chile Pilot
In 2020-2021, a tertiary education provider in Chile ran a pilot program using the Code Avengers platform for their learners. The institution had previously struggled with a failure rate of over 30% for their

Foundations of Computer Science course, which covered topics such as the basics of computer programming and web development.

A number of campuses were given access to the platform as well as guidance on which courses learners should complete to support their coursework. A threshold of 2000 points was established, representing the Code Avengers courses best aligned with a particular segment of their coursework. Of the students that met or exceeded the 2000-point threshold, 86% passed the course. This meant a 14% failure rate–less than half of the previous 30+% failure rate. There was a strong correlation, therefore, between learner engagement with the Code Avengers platform, and pass rates.

Chicago Case Study

In 2012, the Chicago Public Schools recognized a discrepancy: the fastest-growing job sector, Information Technology, was predominantly white and overwhelmingly male. They were determined to shift that balance.

When the Chicago Public Schools began to pilot an "Exploring Computer Science" course several years ago, the staff at Whitney Young High School jumped on board. Their primary goal was to increase diversity in the computer science field - more women, and more people of color. The innovation took hold and before long, the staff at Whitney Young determined that they needed more - more content, more lesson plans, and more assessments. They wanted to take computer science instruction to the next level.

One of the major reasons why CS gained so much traction at Whitney Young was due to them adopting Code Avengers as a tool. Code Avengers helps break down the stereotypes around Computer Science. Students at Whitney Young saw that it was not just the stereotypical math-and-science-minded individuals who can succeed in Computer Science. Many of the students who took the introductory Computer Science course gained confidence and continued on to explore advanced coursework including AP level Computer Science courses. This opened up more opportunities for students to challenge themselves with college-level Computer Science instruction.

Testimonials

The University of Auckland:

> "Our course is a core component of the Bachelor's Degree of Commerce, with well over 2,000 students enrolled each year. Providing personal training in coding, along with helpful evaluation and feedback for each student in such a course is extremely difficult, if not impossible. Since 2014, codeavengers.com has been supporting our course with its outstanding products, including suitable coding lessons for each individual student and fantastic technical support. The teaching team has been receiving positive feedback from the class about the learning environment on the website, and how it encourages students to further explore different aspects of programming. On the other hand, the student evaluation functionalities on codeavengers.com made it so much easier to mark and monitor students' progress. Thanks to the team at codeavengers.com, our faculty now has a course that can effectively and efficiently introduce large groups of students to the amazing world of programming."

David (Head of Department, Tawa College, NZ):

> "**What students like:** We have received positive feedback about the online courses. Students can learn in their own time at their own pace. Nice simple interface, not too much text for instructions with help functions. Also, the level of learning aligns really well with what is expected at years 11, 12, and 13 from NZQA [New Zealand Qualifications Authority].
>
> **What teachers like:** We have appreciated the fact that the courses are at a good level and align with the NZQA Assessments that we set for our students. Examples; Python 1, 2, and 3 all scaffold nicely with students completing the online course for homework during the Term, then as teachers we recap the learning on CA in class so students are ready for their final NZQA / NCEA [National Certificate in Educational Achievement] Assessment."

Tatiana (Independent Learner):

> “It is my first time using code avengers. It is a really well-structured page and I enjoy it. I have never seen a code-teaching website that is this good. The rest that I have seen are mediocre at best. Although Khan Academy had some interesting stuff. But I 100% prefer code avengers and I will continue to learn using it.”

Aimee (Teacher, Sage School, MA):

> “Code Avengers is the perfect platform for kids to gain fluency in programming languages. We serve gifted students so the option for kids to independently advance ahead in lessons is crucial."

Whitney Young High School (Chicago, IL)

> “Student work is self-paced, monitored easily through the progress monitoring dashboard, and teachers can use that data to support student learning with one-on-one or small group instruction, pairing students for peer coaching, or offering additional challenges for those who master content quickly.
>
> Code Avengers delivers everything necessary for a school to equip students with the digital technology skills that will start them on the path to be primed to apply for (and land) the computer-science-based jobs of today and tomorrow: a wide range of Computer Science content, pedagogically sound ready-to-use lesson plans, solid teacher support, progress monitoring, engaging and scaffolded lesson plans, and on-target assessments.”

Andrew Mauer-Oats (Whitney Young High School):

> “Code Avengers gives my advanced students the chance to run on ahead; nothing holds them back from learning all that they wish to learn and the platform is loaded with tons of content giving my students the chance to learn HTML, CSS, or Python.”

Technical Specifications

Operating Systems

Code Avengers runs in the browser, so it is compatible with all common operating systems including:

- Windows
- Chrome
- iOS (iPads)
- MacOS

Devices

Code Avengers can be used on a variety of devices, as long as the screen resolution is at least 1024px x 720px, including:

- Chromebooks
- iPads
- Desktop PCs
- Laptops

Coding courses generally require a device with a keyboard, so Chromebooks, laptops, and desktops are recommended for the best experience with HTML/CSS, web development, and programming-type courses.

Web Browsers

Code Avengers works best in Google Chrome, for the most optimized experience. Code Avengers is compatible with these browsers, and others built on the same technology:

- Google Chrome
- Firefox
- Microsoft Edge
- Safari

Learning Management Systems

Code Avengers is fully compatible with both GSuite/Google Classroom and Canvas LMS’s. We are compatible with other LMSs of a similar caliber and offer teacher tools that make it easy to share particular courses, lessons, and other URLs within other LMS environments.

Programming Languages Taught

Code Avengers courses cover the following programming languages:

- Blockly
- JavaScript including the PaperScript and jQuery libraries
- Python

In addition, we also cover HTML and CSS.

Hardware Requirements

Apart from a compatible device, there are no specific hardware requirements for Code Avengers content. In the not-too-distant future, we are looking at developing more resources to support hardware-related learning in topics such as computer systems, and electronics-related using equipment such as Microbits. Wherever possible, however, we endeavor to offer virtual alternatives in order to support equity in access as not all teachers and learners have access to hardware such as this.

Additional Information

Teacher Friendliness & Usability

This has largely been addressed throughout this report, but just to reinforce, our materials are developed by teachers, for teachers, and have been used extensively by students. They are continually polished in response to feedback, further improving user-friendliness.

Tools & Materials Required or Recommended

Site licenses are required to access most Code Avengers content and teacher resources. Aside from that, nothing additional is required, as all teaching guides, lesson plans, quizzes, projects, assessment resources, and so on are included with the licenses.

Additional Tools/Materials Available

In addition to the online course content, additional resources as outlined in earlier sections of this report are available. These are all included with the site license, so are not required to be purchased separately.

Free Resources

Code Avengers offers a small number of free courses as tasters. These include our most recent Hour of Code courses, as well as a selection of other courses that is revised from time to time. Free courses also come with teaching guides, and unplugged activities to support further learning.

CODE AVENGERS

CAMBRIDGE IGCSE CURRICULUM

COMPUTER SCIENCE ALIGNMENT

CAMBRIDGE IGCSE CURRICULUM
COMPUTER SCIENCE ALIGNMENT

An IGCSE can be taught over 1 or 2 school years. The recommended guided teaching time is 130 hours. Students take 2 written papers. Paper 1 covers topics 1 - 6 and Paper 2 covers topics 7 - 10. Past Papers can be found here for reference.

Topics:

1. Data Representation
2. Data Transmission
3. Hardware
4. Software
5. The Internet & Its Users
6. Automated Systems
7. Algorithmic Design & Problem Solving
8. Programming
9. Databases
10. Boolean Logic

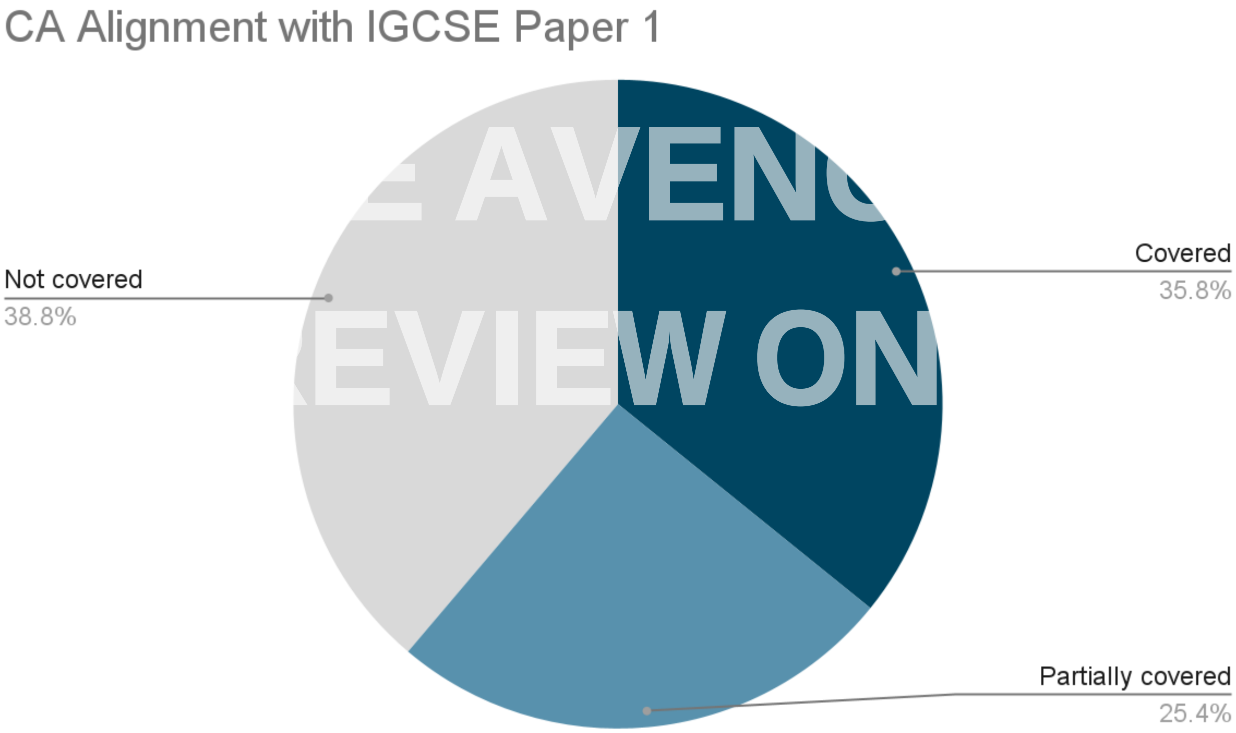
Topics typically contain 3 or more sub-topics. There are 37 sub topics in total. There are 67 learning outcomes (LOs) in Paper 1, and 29 LOs in Paper 2, making 96 LOs total. The following sections will illustrate the coverage of these standards by Code Avengers content. Appendices are included with further background and detail.

Please see next page for alignments by paper

Alignment by Paper

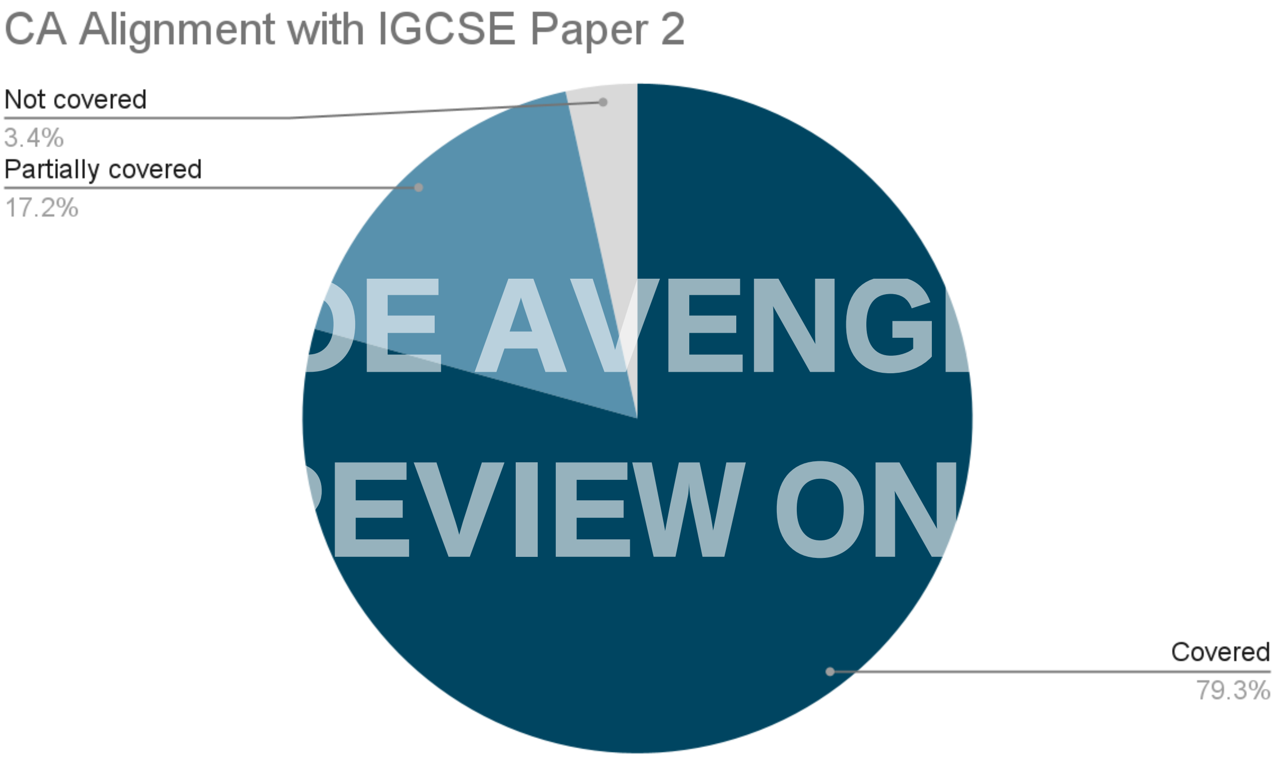
Paper 1

Topic	Yes	Partial	No
1. Data Representation	9	2	2
2. Data Transmission	1	2	6
3. Hardware	3	2	13
4. Software	3	2	4
5. The Internet & Its Users	7	3	0
6. Automated Systems	1	6	1
Totals (out of 67 LOs)	**24**	**17**	**26**



Paper 2

Topic	Yes	Partial	No
1. Algorithmic Design & Problem Solving	8	1	0
2. Programming	13	0	0
3. Databases	0	3	1
4. Boolean Logic	2	1	0
Totals (out of 29 LOs)	**23**	**5**	**1**



Appendix 1: Background & Detail

Across the two papers, there are 96 learning outcomes. Code Avengers content covers 47, partially covers 22, and doesn't cover 27. Those which are not covered are primarily in Paper 1. There is only one LO in Paper 2 that is not covered at all.

More details about the nature and format of each paper are provided below. The full list of LOs for each paper and the corresponding alignment can be found on the following pages.

An additional appendix is provided with more information about the Code Avengers platform and the support it provides for teachers and students.

Paper 1 – Computer Systems

Written paper, 1 hour 45 minutes, 75 marks

- This question paper consists of short-answer and structured questions set on Topics 1–6 of the subject content. All questions are compulsory, and candidates answer on the question paper.
- This paper assesses all assessment objectives, AO1, AO2 and AO3, and assesses the full grade range, A* to G. This paper is externally assessed.
- Calculators are not allowed in this examination.

Paper 2 – Algorithms, Programming and Logic

Written paper, 1 hour 45 minutes, 75 marks

- This question paper consists of short-answer and structured questions set on Topics 7–10 of the subject content. All questions are compulsory, and candidates answer on the question paper.
- The questions require candidates to have practical programming experience.
- Knowledge of programming language syntax is not examined; in all cases the logic is more important than the syntax.
- This paper assesses all assessment objectives, AO1, AO2 and AO3, and assesses the full grade range, A* to G. This paper is externally assessed. Calculators are not allowed in this examination.

PAPER 1		
1.	**Data representation**	
1.1.	Number Systems	
1.1.1.	Understand how and why computers use binary to represent all forms of data	🟩
1.1.2.	Understand the denary, binary and hexadecimal number systems	🟩
1.1.3.	Understand how and why hexadecimal is used as a beneficial method of data representation	🟩
1.1.4.	Add two positive 8-bit binary integers, Understand the concept of overflow and why it occurs in binary addition	🟩
1.1.5.	Perform a logical binary shift on a positive 8-bit binary integer and understand the effect this has on the positive binary integer	🟥
1.1.6.	Use two's complement to represent positive and negative 8-bit binary integers	🟥
1.2.	Text, Sound and images	
1.2.1.	Understand how and why a computer represents text and the use of character sets, including American standard code for information interchange (ASCII) and Unicode	🟩
1.2.2.	Understand how and why a computer represents sound, including the effects of the sample rate and sample resolution	🟩
1.2.3.	Understand how and why a computer represents an image, including the effects of the resolution and colour depth	🟩
1.3.	Data storage and compression	
1.3.1.	Understand how data storage is measured	🟩
1.3.2.	Calculate the file size of an image file and a sound file, using information given	🟨
1.3.3.	Understand the purpose of and need for data compression	🟩
1.3.4.	Understand how files are compressed using lossy and lossless compression methods	🟨
2.	**Data Transmission**	

2.1.	Types and methods of data transmission	
2.1.1.	Understand that data is broken down into packets to be transmitted/Describe the structure of a packet/Describe the process of packet switching	🟨
2.1.2.	Describe how data is transmitted from one device to another using different methods of data transmission/Explain the suitability of each method of data transmission, for a given scenario	🟩
2.1.3.	Understand the universal serial bus (USB) interface and explain how it is used to transmit data	🟥
2.2.	Methods of error detection	
2.2.1.	Understand the need to check for errors after data transmission and how these errors can occur	🟥
2.2.2.	Describe the processes involved in each of the following error detection methods for detecting errors in data after transmission: parity check (odd and even), checksum and echo check	🟥
2.2.3.	Describe how a check digit is used to detect errors in data entry and identify examples of when a check digit is used, including international standard book numbers (ISBN) and bar codes	🟥
2.2.4.	Describe how an automatic repeat query (ARQ) can be used to establish that data is received without error	🟥
2.3.	Encryption	
2.3.1.	Understand the need for and purpose of encryption when transmitting data	🟨
2.3.2.	Understand how data is encrypted using symmetric and asymmetric encryption	🟨
3.	**Hardware**	
3.1.	Computer architecture	
3.1.1.	Understand the role of the central processing unit (CPU) in a computer/Understand what is meant by a microprocessor	🟥
3.1.2.	Understand the purpose of the components in a CPU, in a computer that has a Von Neumann architecture/Describe the process of the fetch–decode–execute cycle including the role of each component in the process	🟥

3.1.3.	Understand what is meant by a core, cache and clock in a CPU and explain how they can affect the performance of a CPU	🟥
3.1.4.	Understand the purpose and use of an instruction set for a CPU	🟥
3.1.5.	Describe the purpose and characteristics of an embedded system and identify devices in which they are commonly used	🟥
3.2.	Input and Output Devices	
3.2.1.	Understand what is meant by an input device and why it is required	🟥
3.2.2.	Understand what is meant by an output device and why it is required	🟥
3.2.3.	Understand what is meant by a sensor and the purposes of sensors / Identify the type of data captured by each sensor and understand when each sensor would be used, including selecting the most suitable sensor for a given context	🟥
3.3.	Data Storage	
3.3.1.	Understand what is meant by primary storage	🟥
3.3.2.	Understand what is meant by secondary storage	🟥
3.3.3.	Describe the operation of magnetic, optical and solid-state (flash memory) storage and give examples of each	🟨
3.3.4.	Describe what is meant by virtual memory, how it is created and used and why it is necessary	🟥
3.3.5.	Understand what is meant by cloud storage	🟩
3.3.6.	Explain the advantages and disadvantages of storing data on the cloud in comparison to storing it locally	🟨
3.4.	Network Hardware	
3.4.1.	Understand that a computer needs a network interface card (NIC) to access a network	🟥
3.4.2.	Understand what is meant by and the purpose of a media access control (MAC) address, including its structure	🟩
3.4.3.	(a) Understand what is meant by and the purpose of an internet protocol (IP) address/(b) Understand that there are different types of IP address	🟩
3.4.4.	Describe the role of a router in a network	🟩
4.	**Software**	

4.1.	Types of software and interrupts	
4.1.1.	Describe the difference between system software and application software and provide examples of each	🟥
4.1.2.	Describe the role and basic functions of an operating system	🟥
4.1.3.	Understand how hardware, firmware and an operating system are required to run applications software	🟥
4.1.4.	Describe the role and operation of interrupts	🟥
4.2.	Types of programming language, translators and IDEs	
4.2.1.	Explain what is meant by a high-level language and a low-level language, including the advantages and disadvantages of each	🟩
4.2.2.	Understand that assembly language is a form of low-level language that uses mnemonics, and that an assembler is needed to translate an assembly language program into machine code	🟨
4.2.3.	Describe the operation of a compiler and an interpreter, including how high-level language is translated by each and how errors are reported	🟩
4.2.4.	Explain the advantages and disadvantages of a compiler and an interpreter	🟩
4.2.5.	Explain the role of an IDE in writing program code and the common functions IDEs provide	🟨
5.	**The internet and its users**	
5.1.	The internet and the world wide web	
5.1.1.	Understand the difference between the internet and the world wide web	🟩
5.1.2.	Describe the purpose and operation of hypertext transfer protocol (HTTP) and hypertext transfer protocol secure (HTTPS)	🟩
5.1.3.	Describe the purpose and operation of hypertext transfer protocol (HTTP) and hypertext transfer protocol secure (HTTPS)	🟩
5.1.4.	Explain the purpose and functions of a web browser	🟩
5.1.5.	Describe how web pages are located, retrieved and displayed on a device when a user enters a URL	🟩

5.1.6.	Explain what is meant by cookies and how they are used, including session cookies and persistent cookies	🟨
5.2.	Digital Currency	
5.2.1.	Understand the concept of a digital currency and how digital currencies are used	🟩
5.2.2.	Understand the process of blockchain and how it is used to track digital currency transactions	🟩
5.3.	Cybersecurity	
5.3.1.	Describe the processes involved in, and the aim of carrying out, a range of cyber security threats	🟨
5.3.2.	Explain how a range of solutions are used to help keep data safe from security threats	🟨
6.	**Automated and emerging technologies**	
6.1.	Automated Systems	
6.1.1.	Describe how sensors, microprocessors and actuators can be used in collaboration to create automated systems	🟥
6.1.2.	Describe the advantages and disadvantages of an automated system used for a given scenario	🟨
6.2.	Robotics	
6.2.1.	Understand what is meant by robotics	🟨
6.2.2.	Describe the characteristics of a robot	🟨
6.2.3.	Understand the roles that robots can perform and describe the advantages and disadvantages of their use	🟨
6.3.	Artificial Intelligence	
6.3.1.	Understand what is meant by artificial intelligence (AI)	🟨
6.3.2.	Describe the main characteristics of AI as the collection of data and the rules for using that data, the ability to reason, and can include the ability to learn and adapt	🟨
6.3.3.	Explain the basic operation and components of AI systems to simulate intelligent behavior	🟨

PAPER 2		
7.	**Algorithmic design and problem solving**	
7.1.	Understand the program development life cycle, limited to: analysis, design, coding and testing	🟩
7.2.	Understand that every computer system is made up of sub-systems, which are made up of further sub-systems/Understand how a problem can be decomposed into its component parts/ Use different methods to design and construct a solution to a problem	🟩
7.3.	Explain the purpose of a given algorithm	🟩
7.4.	Understand standard methods of solution	🟨
7.5.	Understand the need for validation checks to be made on input data and the different types of validation check/Understand the need for verification checks to be made on input data and the different types of verification check	🟩
7.6.	Suggest and apply suitable test data	🟩
7.7.	Complete a trace table to document a dry-run of an algorithm	🟩
7.8.	Identify errors in given algorithms and suggest ways of correcting these errors	🟩
7.9.	Write and amend algorithms for given problems or scenarios, using: pseudocode, program code and flowcharts	🟩
8.	**Programming**	
8.1.	Programming concepts	
8.1.1.	Declare and use variables and constants	🟩
8.1.2.	Understand and use the basic data types	🟩
8.1.3.	Understand and use input and output	🟩
8.1.4.	Understand and use the concept of sequence/selection/iteration/Understand and use the concepts of	🟩

	totalling and counting/ string handling/Understand and use arithmetic, logical and Boolean operators	🟩
8.1.5.	Understand and use nested statements	🟩
8.1.6.	Understand what is meant by procedures, functions and parameters/Define and use procedures and functions, with or without parameters/Understand and use local and global variables	🟩
8.1.7.	Understand and use library routines	🟩
8.1.8.	Understand how to create a maintainable program	🟩
8.2.	Arrays	
8.2.1.	Declare and use one-dimensional (1D) and two-dimensional (2D) arrays	🟩
8.2.2.	Understand the use of arrays	🟩
8.2.3.	Write values into and read values from an array using iteration	🟩
8.3.	File handling	
8.3.1.	Understand the purpose of storing data in a file to be used by a program	🟩
8.3.2.	Open, close and use a file for reading and writing	🟩
9.	**Databases**	
9.1.	Define a single-table database from given data storage requirements	🟨
9.2.	Suggest suitable basic data types	🟨
9.3.	Understand the purpose of a primary key and identify a suitable primary key for a given database table	🟨
9.4.	Read, understand and complete structured query language (SQL) scripts to query data stored in a single database table	🟥
10.	**Boolean Logic**	
10.1.	Identify and use the standard symbols for logic gates	🟩
10.2.	Define and understand the functions of the logic gates	🟩
10.3.	(a) Use logic gates to create given logic circuits from a:problem statement/logic expression/truth table (b) Complete a truth table from a:problem statement/logic expression/logic circuit	🟨

Appendix 2: Code Avengers Deep Dive

Why Code Avengers?

Background

Code Avengers is the commercial realization of the results of a prolonged and extensive Ph.D. research project that identified 29 core elements required to successfully teach coding. It is a comprehensive online education platform (https://codeavengers.com) that effectively caters to educational institutes and individual learners. Our platform provides interactive, gamified courses that teach computer programming, web development, computational thinking, digital skills, design, and more, through an engaging and effective, "learn-by-doing" approach.

Instruction & Support

Our language vocabulary is tailored to support ESOL learners, including strategies like having both written and spoken instructions and limiting non-jargon vocabulary to 2 thousand most common English words. As a web-based platform, screen reading software makes our courses accessible and all tasks work similarly and have a predictable layout, with audio available for most courses to help reduce literacy barriers.

Our Professional Development courses help teachers develop inclusive teaching practices and strategies to support the learning and achievement of diverse learners with specific needs using digital technologies. Through our support and contact functionality, educators have access to a team of PD facilitators and content developers who support them in a variety of ways, such as assisting them in curriculum development, using our platform to plan individualized instruction, and offering hybrid/inclusive learning opportunities.

Student Engagement

Our course developers are qualified, experienced teachers, who are experts in both the content and pedagogy of Computer Science (CS). They have been instrumental in creating quality content that separates us from other platforms by effectively engaging learners from a diverse range of backgrounds. Our scaffolded learning methods build students' skills gradually which means they are continuously challenged and engaged in the learning process.

At Code Avengers, our whole business is focused on providing innovative, future-focused CS education that empowers all learners to develop the digital skills they need to be the creators and innovators of the future. Our course developers have expertise in inclusiveness and diversity, and Code Avengers courses are therefore designed with culturally responsive learning in mind. Because our team is made up of a diverse range of people with different cultures, genders, and experiences, we are able to provide diverse learners with relatable learning contexts and meaningful activities.

We ensure underserved populations can see themselves reflected within the industry and learning environment. Our courses have a diverse cast of characters from varied cultures, countries, and families. At Code Avengers, we support and encourage people from all walks of life to get excited about being creators and innovators in the digital space. The majority of our content is available in English and Spanish, with a growing library of content in German, French, Japanese, Maori, Dutch, Portuguese, Russian, Chinese/Mandarin, and other languages planned as well. This provides additional support for English Language Learners. Code Avengers has been used by over 2 million students worldwide.

Promoting Learning

At its core, CS is about solving problems. Digital outcomes, products, and services are developed in order to find innovative solutions to problems and opportunities we face. That's why at Code Avengers, we integrate problem-solving skills into every learning resource.

Divergent thinking is supported in courses that require students to generate original ideas and examine a variety of viable solutions. Students can investigate tasks in a 'non-linear' manner using logic puzzles that explicitly teach decision-making and problem-solving processes on our platform. We aim to create an environment in which students can gain firsthand experience with the concepts of optimization, redundancy, and reliability, as well as their implications for technological system design, development, and maintenance.

Students are empowered to take an active role in choosing the courses that they complete. They have access to a whole course library, with filters that can help them select a pathway. They can complete a course from beginning to end, or jump to a specific lesson. The courses focus on students creating code to demonstrate their understanding of a concept. This means that they are not just watching a video or passively learning but actively engaging in the learning material to create and modify work. Our teacher training emphasizes strategies like "Pair Programming", "Try 3 Then Me", and the importance of choice and space to tinker. Our learning content complements these approaches.

Individualized Learning

Teachers love the self-paced Code Avengers curriculum as it encourages and enables students to be confident, independent learners. Our courses are perfect for students of any experience level and have many tools available for students with diverse learning needs. When designing courses, we use a variety of strategies to support students and ensure learning is accessible.

Students enjoy the Code Avengers platform because of all the features it offers including immediate feedback, helpful videos, pop-up vocabulary, hints, a built-in integrated development environment (IDE), achievement tracking, and other resources. All tasks are self-graded with real-time progress tracking provided on the teacher dashboard and class data can be exported to a spreadsheet at any time. Teachers can maintain control of class direction by assigning specific modules to class groups or individual students and simultaneously manage class differentiation needs.

Not only can our content be tailored to our students, but our teachers also benefit from the platform's adaptability. Teachers can learn, explore, and upskill alongside their students without having to be experts. Code Avengers makes it easy for teachers with any amount of experience to facilitate effective CS learning in their classrooms. It is a high priority of ours to provide teachers with a suite of resources to support the delivery and monitoring of student learning, including live student tracking, analytics, lesson plans, quizzes, offline activities, and more.

Collaboration

Unplugged activities aimed at helping students think critically and strategically are available to support many of the courses. Students take on specific roles typical in a development cycle to contribute to a project as a team and are taught to manage their time and resources to produce digital artifacts. There are multiple opportunities to collaborate on Code Avengers. Our Pro games and website are shareable and we have JS and HTML/CSS IDEs as practice spaces that allow sharing your code and the ability to download work.

We also provide support and guidance for teachers that includes effective pedagogy and teaching strategies to promote collaboration in Computer Science learning.

Collaboration is fundamental to our internal ethos. We encourage learners and teachers to provide feedback and make suggestions for how to improve our platform, and we actively seek both perspectives when identifying new resources for development.

Real-World Applications

Our courses offer authentic learning opportunities that propose real-world challenges. This allows students to apply their understanding of CS concepts through memorable and compelling applications. Students are expected to use higher-order cognitive processes to analyze problems using a range of skills, such as detail-oriented coding, teamwork, and big-picture, long-term thinking.

Our Junior courses are story-based, and the learning happens within relevant real-world scenarios. At the Pro (senior) level, students write code to build real-world programs and apps that have actual everyday applications.

The impacts of computing are an important topic addressed in many of our courses. Students design and iteratively develop computational artifacts for practical purposes, personal expression, and to address societal issues. They are required to critically evaluate computational artifacts and consider both their beneficial and harmful effects on society. At higher levels, they theorize about the future of the computing innovations that have altered our culture and discuss the legal and ethical implications of these innovations.

Career & Technical Education

Students transition to learning industry-standard programming languages, such as JavaScript and Python, and web languages such as HTML5 and CSS3, or build their theoretical knowledge with our Computer Science and Design courses. These are some of the most in-demand languages in the tech sector, whilst also being very beginner-friendly for both students and teachers. This means learners are given an excellent foundation to move into entry-level/internship-type roles, or on to further study.

Cross-Curricular Connections

Cross-curricular links help learners to make connections with a range of other domains including further STEM topics, social sciences, and the arts. As a company that is already well established in the field of EdTech, Code Avengers is especially well placed to continue to adapt our product to enhance cross-curricular learning. This positions us well to support schools and teachers with the introduction of CS learning as curricula begin to roll out across various states of the USA. We understand that teachers sometimes need to work CS into their classrooms in intelligent ways, so our courses are built to integrate into mainstream learning and connect concepts across the curriculum. When developing new content, cross-curricular connections are a fundamental part of the planning process. Some examples of cross-curricular links found in our existing content are:

- Design theory e.g. color theory.
- History e.g. the history of numbering systems, the evolution of agriculture with AI and agritech.
- Social sciences e.g. the ethics and social impacts of current and future technology.
- Language & literacy e.g. writing and literature-themed activities and writing prompts, procedural writing, constructing texts for specific audiences.
- Science e.g. the physics of light and sound.
- Math e.g. geometry, coordinates, statistics, financial literacy, and more.

Assessment

The Code Avengers learning platform has a number of assessment opportunities available for learners.

Automatic Checking and Validation

Every task on the platform is assessed through automatic checking and validation, to ensure the learner has completed the task properly. This means learners are unable to progress to the next step of their learning without demonstrating an understanding of the step they are currently on. Hints and solutions are available to support learners so that they can self-assess, and practice self-reflection to monitor their own learning. When students use these features, assessment data is generated and displayed as a color-coded progress report in the teacher dashboard view. This means that formative assessment happens naturally as a part of the learning process, and teachers can use this data to identify learners in need of further support without any additional assessment or evaluation work needed on their part.

Quizzes

Our platform also has built-in quizzes that support the recapping and consolidation of information learned in the courses. Lesson plans contain Kahoot quizzes, Google Forms, and other tasks that can also be used to assess student learning for formative or summative purposes.

Projects

Shorter projects that require similar skills are connected to many courses. These have less scaffolding in the instructions in order to support the application of skills learned in the courses. This allows students to try to solve problems more independently and provides teachers with some insight into how well learners have grasped core concepts.

Our step-by-step lesson plans are organized into units that culminate in a project where learners apply the skills they have gained. During this process, learners co-construct and frequently refer to a rubric based on success criteria that they have identified. This gives them the skills they need to be able to feed forward in both self and peer assessment.

Supporting materials

Additional resources such as unplugged activities, workshops, and more, are available to allow for a range of ways to develop and assess learners' understanding of the concepts in the courses.

Standard alignment

Assessment is embedded within all Code Avengers courses. This means that the same coverage, as shown in the curriculum alignment above, applies for both learning content and assessment. The additional resources such as quizzes, projects, and lesson plans that are wrapped around the course content offer opportunities for learning to be assessed in multiple ways.

Performance descriptors

All of our courses come with teaching guides that include detailed breakdowns of the curriculum links, learning outcomes, and task objectives for every lesson and task in the course. These teaching guides also include key vocabulary and definitions so that any misconceptions or misunderstandings can be addressed before teaching and learning are undertaken.

Our growing library of fully guided, step-by-step unit and lesson plans also includes clear curriculum links, learning outcomes, and project options for summatively assessing learning from the unit.

This makes it clear to teachers what learning is being assessed in each lesson, course, and task, and what learners would be expected to be able to do/perform upon completion of the lesson or course.

Assessment Equity

Throughout Code Avengers courses, real-time assessment is carried out using a wide variety of activity and widget types, with all concepts assessed more than once throughout a course, using different learning modes. Courses are designed with diverse learning styles in mind, which means learners get multiple opportunities to apply their learning, in a way that suits them.

Audio is provided throughout many of the courses, particularly those aimed at younger learners, to support those who may struggle with literacy demands, including English Language Learners. Videos are provided in many courses for older learners as well.

Our step-by-step lesson plans offer guidance for differentiated learning in order to support a diverse range of learners, from those who need additional support, to gifted learners. The self-paced nature of Code Avengers, and the ability to assign different courses and modules to various learners, further support teachers to deliver differentiated programs of learning and assessment.

Professional Development

No experience or specific professional development (PD) training is required in order to successfully use Code Avengers to support Computer Science learning in the classroom. However, we do offer a range of resources to support teachers to get the best out of using Code Avengers with their learners.

Our courses are scaffolded to support educators and learners of all experience levels. Teachers can complete courses themselves to upskill or learn alongside students collaboratively. The rich set of resources, with interactive tasks and activities, is ideal for teachers to learn computer science concepts, programming languages, and other applications of real-world skills.

In addition to the full course library, EDU licenses provide access to our interactive Professional Development courses, covering topics such as computer science concepts, collaborative teaching strategies and effective pedagogy for computer science, repositioning the teacher from being the expert to learning alongside their students, fostering perseverance and innovation, and culturally responsive pedagogy. This is valuable learning, given the significant number of teachers who come into teaching Computer Science without a background in the subject, and the importance of being able to engage diverse learners.

Code Avengers is an accredited PD provider, with extensive experience delivering in-person and online professional development programs in numerous settings across the globe. We at Code Avengers will always provide support on our platform to our customers, as an included part of our software as a service. We can also provide custom, personal, and expert professional development to any teacher, school, or district that could benefit from it, in-person or online.

We have consistently and effectively delivered high-quality PD since our inception. Teachers' confidence and time are a top priority, and we can provide additional tools beyond our included support to build confidence and save time.

PD Testimonial (Roz, Maihihi School):

> “This was an excellent piece of PD which was relevant and well facilitated. Thank you so much. The presenters were great. For myself, as a non-confident digital user, they were very considerate and understanding in helping me to work through what was to be taught and the best way to support my students with their digital learning. Thank you for giving me the confidence to be a digital learner and a digital teacher.“

Please contact the Code Avengers team to learn more about our current offerings and schedule. More testimonials are included below.

Research & Testimonials

Research & Case Studies

Queensland Pilot

In late 2019, a trial was conducted with Trinity Bay State High School in Manunda, Queensland Australia. It has a demographic of approximately 35% indigenous learners. Having previously had difficulty engaging their learners with digital technologies, the school piloted the Code Avengers platform to explore whether it would improve student engagement and learning outcomes.

The trial was a resounding success, with student engagement significantly improved, particularly for indigenous girls. The pilot report can be read here, summarized in the quote below with key findings and benefits of using Code Avengers:

> “Differentiation of Digital Technologies course content is now truly possible with the Code Avengers platform. We have a very large Special Needs Program at the school with over 150 students identified with special needs/cognitive disorders/physical issues. Many of these students have worked extremely effectively on the Code Avengers platform. This is an area where their various disabilities are of little consequence. Many of the specialist staff that support them have been extremely surprised at the results that the students have achieved using Code Avengers.”

Chile Pilot

In 2020-2021, a tertiary education provider in Chile ran a pilot program using the Code Avengers platform for their learners. The institution had previously struggled with a failure rate of over 30% for their Foundations of Computer Science course, which covered topics such as the basics of computer programming and web development.

A number of campuses were given access to the platform as well as guidance on which courses learners should complete to support their coursework. A threshold of 2000 points was established, representing the Code Avengers courses best aligned with a particular segment of their coursework. Of the students that met or exceeded the 2000-point threshold, 86% passed the course. This meant a 14% failure rate– less than half of the previous 30+% failure rate. There was a strong correlation, therefore, between learner engagement with the Code Avengers platform, and pass rates.

Chicago Case Study

In 2012, the Chicago Public Schools recognized a discrepancy: the fastest-growing job sector, Information Technology, was predominantly white and overwhelmingly male. They were determined to shift that balance.

When the Chicago Public Schools began to pilot an “Exploring Computer Science” course several years ago, the staff at Whitney Young High School jumped on board. Their primary goal was to increase diversity in the computer science field - more women, and more people of color. The innovation took hold and before long, the staff at Whitney Young determined that they needed more - more content, more lesson plans, and more assessments. They wanted to take computer science instruction to the next level.

One of the major reasons why CS gained so much traction at Whitney Young was due to them adopting Code Avengers as a tool. Code Avengers helps break down the stereotypes around Computer Science. Students at Whitney Young saw that it was not just the stereotypical math-and-science-minded individuals who can succeed in Computer Science. Many of the students who took the introductory Computer Science course gained confidence and continued on to explore advanced coursework including AP level Computer Science courses. This opened up more opportunities for students to challenge themselves with college-level Computer Science instruction.

Testimonials

The University of Auckland:

> “Our course is a core component of the Bachelor’s Degree of Commerce, with well over 2,000 students enrolled each year. Providing personal training in coding, along with helpful evaluation and feedback for each student in such a course is extremely difficult, if not impossible. Since 2014, codeavengers.com has been supporting our course with its outstanding products, including suitable coding lessons for each individual student and fantastic technical support. The teaching team has been receiving positive feedback from the class about the learning environment on the website, and how it encourages students to further explore different aspects of programming. On the other hand, the student evaluation functionalities on codeavengers.com made it so much easier to mark and monitor students’ progress. Thanks to the team at codeavengers.com, our faculty now has a course that can effectively and efficiently introduce large groups of students to the amazing world of programming.”

David (Head of Department, Tawa College, NZ):

> "**What students like:** We have received positive feedback about the online courses. Students can learn in their own time at their own pace. Nice simple interface, not too much text for instructions with help functions. Also, the level of learning aligns really well with what is expected at years 11, 12, and 13 from NZQA [New Zealand Qualifications Authority].
>
> **What teachers like:** We have appreciated the fact that the courses are at a good level and align with the NZQA Assessments that we set for our students. Examples; Python 1, 2, and 3 all scaffold nicely with students completing the online course for homework during the Term, then as teachers we recap the learning on CA in class so students are ready for their final NZQA / NCEA [National Certificate in Educational Achievement] Assessment."

Tatiana (Independent Learner):

> “It is my first time using code avengers. It is a really well-structured page and I enjoy it. I have never seen a code-teaching website that is this good. The rest that I have seen are mediocre at best. Although Khan Academy had some interesting stuff. But I 100% prefer code avengers and I will continue to learn using it.”

Aimee (Teacher, Sage School, MA):

> “Code Avengers is the perfect platform for kids to gain fluency in programming languages. We serve gifted students so the option for kids to independently advance ahead in lessons is crucial."

Whitney Young High School (Chicago, IL)

> “Student work is self-paced, monitored easily through the progress monitoring dashboard, and teachers can use that data to support student learning with one-on-one or small group instruction, pairing students for peer coaching, or offering additional challenges for those who master content quickly.
>
> Code Avengers delivers everything necessary for a school to equip students with the digital technology skills that will start them on the path to be primed to apply for (and land) the computer-science-based jobs of today and tomorrow: a wide range of Computer Science content, pedagogically sound ready-to-use lesson plans, solid teacher support, progress monitoring, engaging and scaffolded lesson plans, and on-target assessments.”

Andrew Mauer-Oats (Whitney Young High School):

> “Code Avengers gives my advanced students the chance to run on ahead; nothing holds them back from learning all that they wish to learn and the platform is loaded with tons of content giving my students the chance to learn HTML, CSS, or Python.”

Technical Specifications

Operating Systems

Code Avengers runs in the browser, so it is compatible with all common operating systems including:

- Windows
- Chrome
- iOS (iPads)
- MacOS

Devices

Code Avengers can be used on a variety of devices, as long as the screen resolution is at least 1024px x 720px, including:

- Chromebooks
- iPads
- Desktop PCs
- Laptops

Coding courses generally require a device with a keyboard, so Chromebooks, laptops, and desktops are recommended for the best experience with HTML/CSS, web development, and programming-type courses.

Web Browsers

Code Avengers works best in Google Chrome, for the most optimized experience. Code Avengers is compatible with these browsers, and others built on the same technology:

- Google Chrome
- Firefox
- Microsoft Edge
- Safari

Learning Management Systems

Code Avengers is fully compatible with both GSuite/Google Classroom and Canvas LMS's. We are compatible with other LMSs of a similar caliber and offer teacher tools that make it easy to share particular courses, lessons, and other URLs within other LMS environments.

Programming Languages Taught

Code Avengers courses cover the following programming languages:

- Blockly
- JavaScript including the PaperScript and jQuery libraries
- Python

In addition, we also cover HTML and CSS.

Hardware Requirements

Apart from a compatible device, there are no specific hardware requirements for Code Avengers content. In the not-too-distant future, we are looking at developing more resources to support hardware-related learning in topics such as computer systems, and electronics-related using equipment such as Microbits. Wherever possible, however, we endeavor to offer virtual alternatives in order to support equity in access as not all teachers and learners have access to hardware such as this.

Additional Information

Teacher Friendliness & Usability

This has largely been addressed throughout this report, but just to reinforce, our materials are developed by teachers, for teachers, and have been used extensively by students. They are continually polished in response to feedback, further improving user-friendliness.

Tools & Materials Required or Recommended

Site licenses are required to access most Code Avengers content and teacher resources. Aside from that, nothing additional is required, as all teaching guides, lesson plans, quizzes, projects, assessment resources, and so on are included with the licenses.

Additional Tools/Materials Available

In addition to the online course content, additional resources as outlined in earlier sections of this report are available. These are all included with the site license, so are not required to be purchased separately.

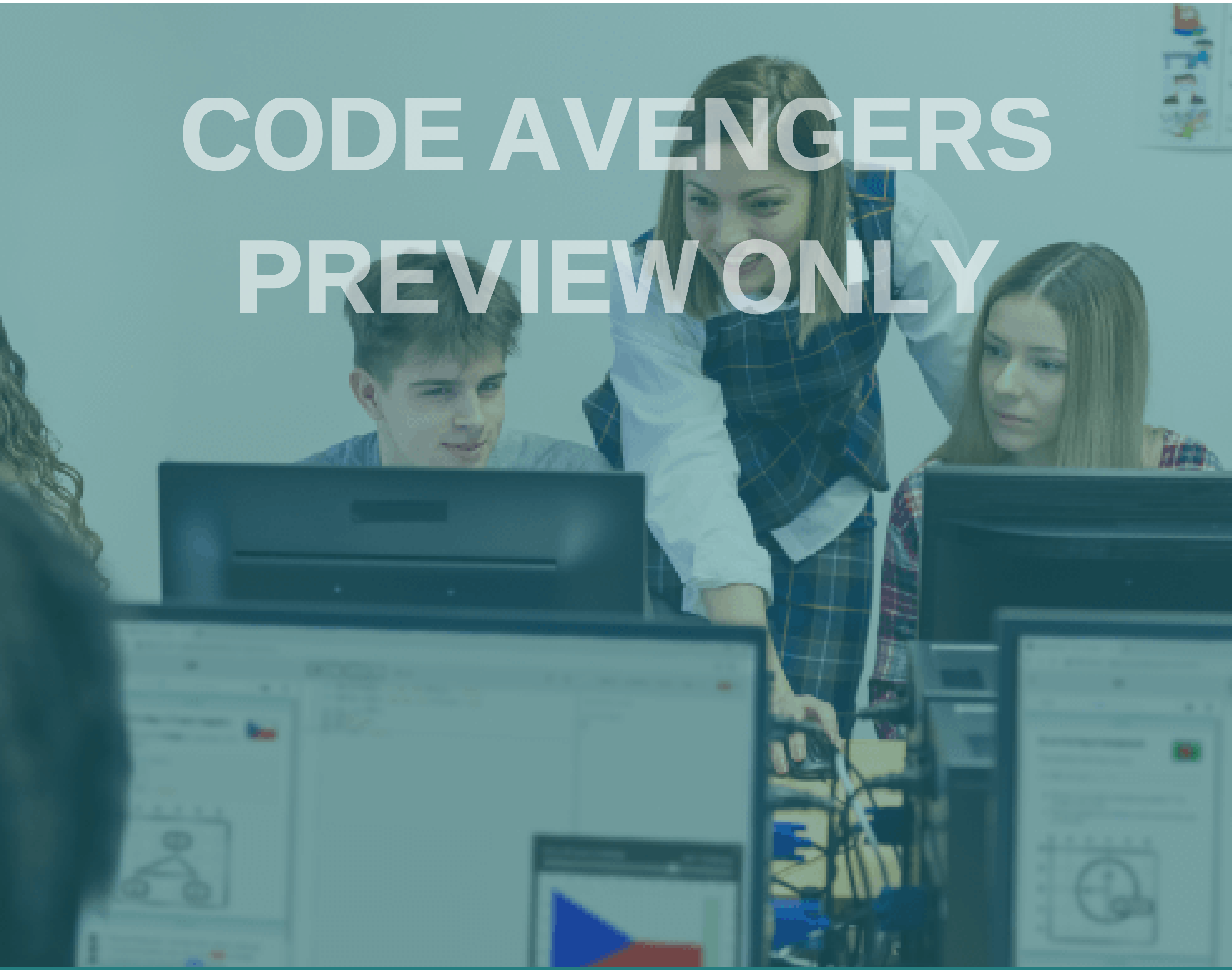
Free Resources

Code Avengers offers a small number of free courses as tasters. These include our most recent Hour of Code courses, as well as a selection of other courses that is revised from time to time. Free courses also come with teaching guides, and unplugged activities to support further learning.

CODE AVENGERS

INTERNATIONAL BACCALAUREATE (IB) CURRICULUM

COMPUTER SCIENCE ALIGNMENT



INTERNATIONAL BACCALAUREATE (IB) CURRICULUM COMPUTER SCIENCE ALIGNMENT

Code Avengers can support student learning across all IB programmes. Our cross-curricular focused Primary courses can support inquiry learning in Primary Years Programme (PYP).

Our Intro and Level 1 courses all teach the skills to create programs, web sites and apps, which students can use to create outcomes for a Digital Design course in the Middle Years Programme (MYP).

For Computer Science in the Diploma Programme (DP) or Career Related Programme (CP) our Level 1 Computer Science courses can support struggling learners develop confidence in select general topics.

IB COMPUTER SCIENCE PATHWAY

Figure 2 shows the IB continuum pathways to DP computer science, design technology and ITGS.

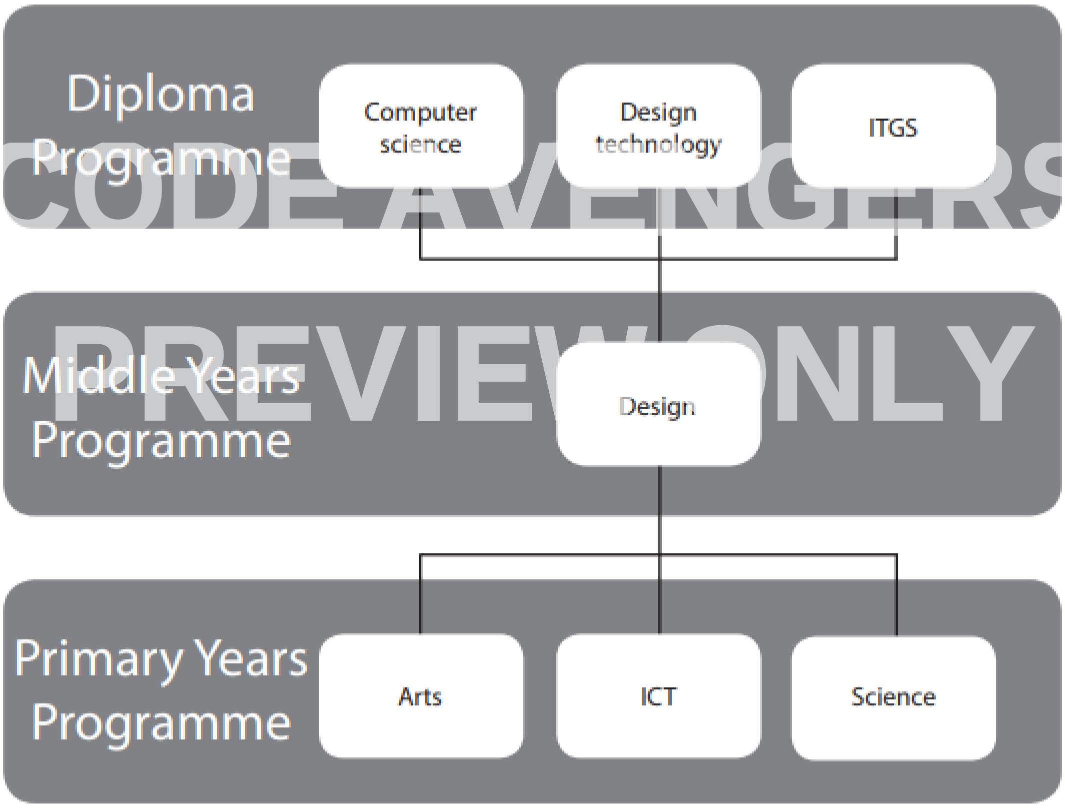


Figure 2
IB continuum pathway to design-related Diploma Programme courses

Primary Years Programme (PYP)

Code Avengers' Junior programmes were created with multiple topics in mind and can form a part of the learning of the inquiry topic, see table.

Note that the numbers indicate the expected age group of the learners.

- Level 1 - Ages 5 to 8 (Years 1-3)
- Level 2 - Ages 7 to 11 (Years 4-6)
- Level 3 - Ages 10 to 13 (Years 7-9–this is above PYP age groups but might provide further extension)

Transdisciplinary Theme - Who We Are	
An inquiry into the nature of the self, beliefs and values; personal, physical, mental, social and spiritual health; human relationships including families, friends, communities and cultures; rights and responsibilities; what it means to be human.	Programming 2: Gear Up For Safety course & Year 5 The Cycle Control Protocol unit - design, physical health, programming, app creation, user safety Year 5 Code Your Own Adventure unit.
Transdisciplinary Theme - Sharing the Planet	
An inquiry into rights and responsibilities in the struggle to share finite resources with other people and with other living things; communities and the relationships within and between them, access to equal opportunities; peace and conflict resolution	Data Representation 1: Carnival Crisis course & Year 1 Data Detectives unit - Math, graph, solving problems/conflict. Computational Thinking 3: Saving Food Avengers - math, algorithms, deconstructing problems, how to formulate and evaluate better solutions.
Transdisciplinary Theme - Where We Are in Place and Time	
An inquiry into orientation in place and time; personal histories; homes and journeys; the discoveries, explorations and migrations of humankind; the relationships between and the interconnectedness of individuals and civilisations, from the local and global perspectives	Data Representation 2: Museum Mystery - communication, history, technology. Data Representation 3: Numbers Through Time - number systems, history, logic, ethics. Computational Thinking 200: Māori New Year - culture, expressing ideas, planting, creating meaning through connecting with objects.
Transdisciplinary Theme - How We Organise Ourselves	
An inquiry into the interconnectedness of human-made systems and communities; the structure and function of organisations; societal	Data Representation 2: Museum Mystery - communication, history, technology.

decision-making; economic activities and their impact on humankind and the environment	Data Representation 3: Numbers Through Time - number systems, history, logic, ethics.
Transdisciplinary Theme - How the World Works	
An inquiry into the natural world and its laws; the interaction between the natural world (physical and biological) and human societies; how humans use their understanding of scientific principles; the impact of scientific and technological advances on society and on the environment	Computational Thinking 1: Camping Adventure course & Year 2 Supernova Sequences unit - patterns, using technology, directions, maps, light pollution, astronomy. Computational Thinking 2: Creature Feature Zoo - maps, nodes, directions, ordering, logic, biology, genetics Programming 0: Code Crazy Creatures - sequence, programming, animation, events. Year 4 Loopy Logic Lab unit.
Transdisciplinary Theme - How We Express Ourselves	
An inquiry into the ways in which we discover and express ideas, feelings, nature, culture, beliefs and values; the ways in which we reflect on, extend and enjoy our creativity; our appreciation of the aesthetic	Programming 1: Stormy Day course & Year 3 Fishful Thinking unit- sequence, biology, story telling, directions. Computational Thinking 200: Māori New Year - culture, expressing ideas, planting, creating meaning through connecting with objects. Programming 3 Larsson Castle Mystery course & Year 6 Castle Coding Capers unit - programming, app design, animation, story telling, shield blazonry. Year 5 Code Your Own Adventure unit.

Middle Years Programme (MYP)

The MYP curriculum framework comprises eight subject groups: Language and Literature (English), Language Acquisition (Second Language), Individual and Societies, Mathematics, Design, Arts, Sciences, Physical Health and Education. It is meant to take 5 years to teach the 400 hours of content… but it can be condensed into less. For example, the Queen Margaret College Middle School teaches MYP in Year 7 – 10, over 4 years.

Like in all NZ middle schools, digital technologies is used as a tool throughout all subjects - researching online, creating digital presentations, and digital citizenship etc. It is completely up to a school to teach the design subject through a digital focus, there is only *some* guidance on topics…

> *"Distinct digital design courses include web design, interactive media design, programming and control, and so on."*[1]

If a school offers **Digital Design** as a subject, they choose/create an inquiry question for each unit of learning across the 5 years (around 11-12 units[2]). Each inquiry question has to build learning in a global context:

- Identities and relationships
- Orientation in space and time
- Personal and cultural expression
- Scientific and technical innovation
- Globalization and sustainability
- Fairness and development

How can CA help? (Intros and Hour of Code Courses/Other Projects)

We can support digital design students in gaining skills towards Assessment Objective C: Creating the solution.

Students can choose a course that will help them gain the skills to create their solution for their design challenges (YR 7-9) and design problems (YR 9-11).[3]

Age appropriate CA topics

Make a basic program	Python 0 or Select lessons in Python 1
Make a basic program (Create a Quiz)	JavaScript 0 or Select lessons in JavaScript 1
Make a basic website	HTML 0 or Select lessons in HTML 1
Make 3 basic apps	Web Design 0 or Select lessons in Web Design 1

[1] Pg16 Design Curriculum guide IBO 2015 https://www.spps.org/site/handlers/filedownload.ashx?moduleinstanceid=38381&dataid=21241&FileName=design_guide_2014.pdf

[2] Whole pdf Design curriculum Renaissance College 2018 (English international school in Hong Kong) https://www.rchk.edu.hk/wp-content/uploads/2018/08/MYP-design-curriculum-display-2018-2019.pdf

[3] Pg38 Design Curriculum guide IBO 2015 https://www.spps.org/site/handlers/filedownload.ashx?moduleinstanceid=38381&dataid=21241&FileName=design_guide_2014.pdf

(Quiz, interactive xylophone, simple ringer app)	

[Note]

Diploma Programme (DP)

In DP Computer Science is in Group 4 Science.
Students can choose to do the Standard Level (SL), or the High Level (HL).

They learn for 2 years and then sit an exam of three papers.

- Paper 1 covers seven topics at HL
    - 1.1 – Systems in Organizations
    - 1.2 – System Design basics
    - 2.1 – Computer organization
    - 3.1 – Networks
    - 4.1 – General Principals
    - 4.2 – Connecting computational thinking and program design
    - 4.3 – Introduction to programming
    - 5.1 – Abstract Data Structures (HL only)
    - 6.1 – Resource Management (HL only)
    - 7.1 – Control (HL only)

- Paper 2 you must select one of four options to answer questions about
    - A—Databases
    - B—Modelling and simulation
    - C—Web Science
    - D—Object Oriented Programming

- Paper 3 is a case study that changes each year.

How does CA support?

Code Avengers offers a range of web, programming, and conceptual courses that support the IB Computer Science syllabus. Below are details of which courses support the learning in each topic.

Topic	CA courses
1.2 – System Design basics	Human interaction with the system Human Computer Interaction 1 Impacts of Computing 600 Networks & Security 600
2.1 – Computer organization	Binary & Logic: Data Representation 3

	Data Representation 400 Data Representation 4 Programming Languages 1
3.1 – Networks	Digital Systems 1 Networks & Security 600 Digital Infrastructure 4 Impacts of Computing 300 (younger target age but a light intro to security)
4.1 – General Principals	Algorithms 1 Software Engineering 1 JavaScript 1-3 and Python 1-3 courses also cover aspects of planning, though they more fully support the programming topic.
4.2 – Connecting computational thinking and program design	Algorithms 1 Software Engineering 1 Programming 3 and Programming 4 - Younger target audience but cover breaking problems down in pseudocode.
4.3 – Introduction to programming	JavaScript 0 (Intro to JS) Python 0 (Intro to JS) Programming Languages 1 JavaScript 1, JavaScript 2 Python 1, Python 2 Year 9 Programming Units: 'Rithms 'n' Hues and Party Parameters
A—Databases	Web Development/Databases: Web Development 1, Web Development 2 and Web Development 3 will help support understanding but does not cover anything specifically
C—Web Science	Digital Systems 1 - Intro to how the web works Networks & Security 600 - Intro to cloud computing and data. Digital Infrastructure 4 - Covers development of the web, protocols, wired vs. wireless networks. Websites HTML 1, HTML 2, and HTML 3 will help support understanding through learning how webpages are built using HTML & CSS. Web Development: Web Development 1, Web Development 2 and Web Development 3 will help support understanding of static vs. dynamic webpages.

D–Object-Oriented Programming	JavaScript 3 Python 3

We would strongly recommend Algorithms 1, Programming Languages 1, Software Engineering 1 and HCI1 to support learning in some general topics.

Strongly aligned courses	Topics they support
Algorithms Level 1 Human and Computer Interaction - Explore system usability and accessibility.	14.1 – General Principals - Thinking procedurally
HCI Level 1 Human and Computer Interaction - Explore system usability and accessibility.	1.2 – System Design basics - Human interaction with the system
Programming Languages 1 Explore features of programming languages and how to translate between high and low level languages.	2.1 – Computer organization - Binary Representation 4.1 – General Principals - Thinking procedurally 4.3 – Introduction to programming
Software Engineering 1 Program Planning	4.1 – General Principals - Thinking ahead 4.2 – Connect CT & program design
Courses with some supporting lessons	**Topics they support**
Computational Thinking 3 JR Cover place value and other fundamentals of common number systems. Lessons 4-5	4.2 – Connect CT & program design
Data Representation 3 JR Cover place value and other fundamentals of common number systems. Lessons 3, 5 Data Representation 400 JR	2.1 – Computer organization - Binary Representation
JavaScript 1 Basic Programming Lessons 3, 5, 13, 19 Python 1 Programming 4	4.3 – Introduction to programming - Use of programming languages
JavaScript 2 Python 2	5.1 – Abstract Data Structures (HL only) - Abstract data structures

Advanced Programming Lessons 32-35	

Specific Code Avengers course support for DP topic points:

Algorithms Level 1 **Human and Computer Interaction - Explore system usability and accessibility.**	**4.1 – General Principals** - **Thinking procedurally** **4.3 – Introduction to programming (HL & SL)** - **Use of programming languages** **IB topic points this lesson supports:**
1 Defining what an algorithm is.	● 4.1.1 Identify the procedure appropriate to solving a problem
2 Building with sequential structures.	● 4.1.2 Evaluate whether the order in which activities are undertaken will result in the required outcome
3 Exploring ways of communicating algorithms.	● 4.1.4 Identify when decision-making is required in a specified situation
4 Building with simple conditional structures.	● 4.1.4 Identify when decision-making is required in a specified situation
5 Building with chained conditional structures.	● 4.1.3 Explain the role of sub-procedures in solving a problem
6 Building with advanced conditional structures.	● 4.1.5 Identify the decisions required for the solution to a specified problem ● 4.1.6 Identify the condition associated with a given decision in a specified problem
7 Building with simple iterative structures.	● 4.3.9 Construct algorithms using loops, branching
8 Building with repeat-while iterative structures.	● 4.3.9 Construct algorithms using loops, branching
9 Building with combined structures.	● 4.3.9 Construct algorithms using loops, branching
10 Building a complex algorithm.	● 4.3.9 Construct algorithms using loops, branching

Computational Thinking 3 JR **Cover place value and other fundamentals of common number systems.**	**4.2 – Connect CT & program design** **IB topic points this lesson supports:**
4: Logic and Problem Solving	• 4.2.8 Deduce the efficiency of an algorithm in the context of its use
5: Iteration and Patterns	• 4.2.9 Determine the number of times a step in an algorithm will be performed for given input data

Data Representation 3 JR **Cover place value and other fundamentals of common number systems.**	**2.1 – Computer organization** **- Binary Representation** **IB topic points this lesson supports:**
3: Decimal - Base 10 and Hexadecimal - Base 16	• 2.1.9 Define the terms: bit, byte, binary, denary/decimal, hexadecimal
5: Binary - Base 2	• 2.1.9 Define the terms: bit, byte, binary, denary/decimal, hexadecimal

HCI Level 1 **Human and Computer Interaction - Explore system usability and accessibility.**	**1.2 – System Design basics** **- Human interaction with the system** **IB topic points this lesson supports:**
1 Understanding what user-interfaces are.	• 1.2.16 Discuss the moral, ethical, social, economic and environmental implications of the interaction between humans and machines
2 Understanding how humans are different to computers.	• 1.2.16 Discuss the moral, ethical, social, economic and environmental implications of the interaction between humans and machines
3 Understanding what usability is.	• 1.2.12 Define the term usability
4 Accessibility: Recognising that different people have different needs.	• 1.2.14 Identify methods that can be used to improve the accessibility of systems
5 Heuristic: Match between the system and the real world	• 1.2.13 Identify a range of usability problems with commonly used digital devices • 1.2.15 Identify a range of usability problems that can occur in a system
6 Heuristic: Visibility of the system status.	• 1.2.13 Identify a range of usability problems with commonly used digital devices

7 Heuristic: User control and freedom.	- 1.2.13 Identify a range of usability problems with commonly used digital devices - 1.2.15 Identify a range of usability problems that can occur in a system
8 Heuristic: Aesthetics and minimalist design.	- 1.2.13 Identify a range of usability problems with commonly used digital devices - 1.2.15 Identify a range of usability problems that can occur in a system
9 Heuristics: Good error messages + Prevent errors.	- 1.2.13 Identify a range of usability problems with commonly used digital devices - 1.2.15 Identify a range of usability problems that can occur in a system
10 Evaluating a user interface.	- 1.2.13 Identify a range of usability problems with commonly used digital devices

JavaScript 1 **Basic Programming**	**4.3 – Introduction to programming** **- Use of programming languages** **IB topic points this lesson supports:**
3: Order of operation math rules	- 4.3.6 Define the terms: variable, constant, operator, object
5: Store data with variables Constants activity	- 4.3.6 Define the terms: variable, constant, operator, object - 4.3.7 Define the operators =, ., <, <=, >, >=, mod, div - 4.3.8 Analyse the use of variables, constants and operators in algorithms
13: True & false	- 4.3.7 Define the operators =, ., <, <=, >, >=, mod, div
19: Combining conditions with && and ||	- 4.3.7 Define the operators =, ., <, <=, >, >=, mod, div

JavaScript 2 **Advanced Programming**	**5.1 – Abstract Data Structures (HL only)** **- Abstract data structures** **IB topic points this lesson supports:**
32. Sorting and printing lists of strings	- 5.1.4 Describe the characteristics of a two-dimensional array
33. Practice creating arrays	- 5.1.5 Construct algorithms using two-dimensional arrays

34. Refactor arrays with graphics	- 5.1.5 Construct algorithms using two-dimensional arrays
35. Arrays practice	- 5.1.5 Construct algorithms using two-dimensional arrays

Programming Languages 1 **Explore features of programming languages and how to translate between high and low level languages.**	**2.1 – Computer organization** **- Binary Representation** **4.1 – General Principals** **- Thinking procedurally** **4.3 – Introduction to programming** **IB topic points this lesson supports:**
1: Different ways of giving instructions: informal, algorithm, program	● General support for 4.1 "Thinking Procedurally"
2: Basic features; output, input, storing, & math	● General support for 4.1 "Thinking Procedurally" ● 4.3.3 Explain the essential features of a computer language
3: Flow control in programming languages	● General support for 4.1 "Thinking Procedurally" ● 4.3.3 Explain the essential features of a computer language
4: Understanding the purpose of syntax	● 4.3.4 Explain the need for higher level languages
5: Adding with circuits	● 2.1.10 Outline the way in which data is represented in the computer. ● 4.3.1 State the fundamental operations of a computer
6: Operators	● 2.1.10 Outline the way in which data is represented in the computer. ● 4.3.1 State the fundamental operations of a computer
7: Bit instructions in a machine language	● 2.1.10 Outline the way in which data is represented in the computer. ● 4.3.1 State the fundamental operations of a computer
8: Compilers	● 4.3.5 Outline the need for a translation process from a higher level language to machine executable code
9: Translating with interpreters	● 4.3.5 Outline the need for a translation process from a higher level language to machine executable code

10: Review programming languages	• General support for 4.1 "Thinking Procedurally"

Software Engineering 1 **Program Planning**	**4.1 – General Principals** **- Thinking ahead** **4.2 – Connect CT & program design** **IB topic points this lesson supports:**
1 Introduction to Planning	• 4.1 General Principals
2 Flowchart Symbols	• 4.1.9 Identify the inputs and outputs required in a solution • 4.2.4 Analyse an algorithm presented as a flow chart
3 Practice Building Flowcharts	• 4.2.4 Analyse an algorithm presented as a flow chart
4 Introduction to Pseudocode	• 4.2.5 Analyse an algorithm presented as pseudocode
5 Practice Using Pseudocode	• 4.2.6 Construct pseudocode to represent an algorithm
6 Planning Conditional Structures	• 4.2.5 Analyse an algorithm presented as pseudocode
7 Iterative Structures and Robustness	• 4.1.10 Identify pre-planning in a suggested problem and solution • 4.1.11 Explain the need for pre-conditions when executing an algorithm • 4.1.12 Outline the pre- and post-conditions to a specified problem • 4.1.13 Identify exceptions that need to be considered in a specified problem solution
8 Flexibility in Planning	• 4.1.10 Identify pre-planning in a suggested problem and solution • 4.1.11 Explain the need for pre-conditions when executing an algorithm • 4.1.12 Outline the pre- and post-conditions to a specified problem • 4.1.13 Identify exceptions that need to be considered in a specified problem solution
9 Planning Functions, Classes and GUIs	• 4.1.10 Identify pre-planning in a suggested problem and solution • 4.1.11 Explain the need for pre-conditions when executing an algorithm • 4.1.12 Outline the pre- and post-conditions to a specified problem

	● 4.1.13 Identify exceptions that need to be considered in a specified problem solution
10 Review What You Have Learned	● 4.1.10 Identify pre-planning in a suggested problem and solution ● 4.1.11 Explain the need for pre-conditions when executing an algorithm ● 4.1.12 Outline the pre- and post-conditions to a specified problem ● 4.1.13 Identify exceptions that need to be considered in a specified problem solution

Career Related Programme (CP)

"The International Baccalaureate® (IB) Career-related Programme (CP) is a three-part educational framework.
It consists of:

- courses from the IB's Diploma Programme (DP)
- the CP core
- career-related studies.

Students complete at least two DP courses in any of that programme's subject groups. DP courses provide and enhance the theoretical underpinnings and academic rigour of the CP.

The CP core components give context to the DP courses and the career-related study, drawing all aspects of the framework together. Through the CP core, students develop personal qualities and professional skills, as well as intellectual habits required for lifelong learning.

Each school chooses the career-related study most suited to local conditions and the needs of its students The career-related study must satisfy IB criteria for accreditation, assessment and quality assurance."[4]

[4] CP Curriculum page, IBO.org. *Accessed 29/1/2019*
https://www.ibo.org/programmes/career-related-programme/curriculum/

Appendix: Code Avengers Deep Dive

Why Code Avengers?

Background

Code Avengers is the commercial realization of the results of a prolonged and extensive Ph.D. research project that identified 29 core elements required to successfully teach coding. It is a comprehensive online education platform (https://codeavengers.com) that effectively caters to educational institutes and individual learners. Our platform provides interactive, gamified courses that teach computer programming, web development, computational thinking, digital skills, design, and more, through an engaging and effective, “learn-by-doing” approach.

Instruction & Support

Our language vocabulary is tailored to support ESOL learners, including strategies like having both written and spoken instructions and limiting non-jargon vocabulary to 2 thousand most common English words. As a web-based platform, screen reading software makes our courses accessible and all tasks work similarly and have a predictable layout, with audio available for most courses to help reduce literacy barriers.

Our Professional Development courses help teachers develop inclusive teaching practices and strategies to support the learning and achievement of diverse learners with specific needs using digital technologies. Through our support and contact functionality, educators have access to a team of PD facilitators and content developers who support them in a variety of ways, such as assisting them in curriculum development, using our platform to plan individualized instruction, and offering hybrid/ inclusive learning opportunities.

Student Engagement

Our course developers are qualified, experienced teachers, who are experts in both the content and pedagogy of Computer Science (CS). They have been instrumental in creating quality content that separates us from other platforms by effectively engaging learners from a diverse range of backgrounds. Our scaffolded learning methods build students’ skills gradually which means they are continuously challenged and engaged in the learning process.

At Code Avengers, our whole business is focused on providing innovative, future-focused CS education that empowers all learners to develop the digital skills they need to be the creators and innovators of the future. Our course developers have expertise in inclusiveness and diversity, and Code Avengers courses are therefore designed with culturally responsive learning in mind. Because our team is made up of a diverse range of people with different cultures, genders, and experiences, we are able to provide diverse learners with relatable learning contexts and meaningful activities.

We ensure underserved populations can see themselves reflected within the industry and learning environment. Our courses have a diverse cast of characters from varied cultures, countries, and families. At Code Avengers, we support and encourage people from all walks of life to get excited about being creators and innovators in the digital space. The majority of our content is available in English and Spanish, with a growing library of content in German, French, Japanese, Maori, Dutch, Portuguese, Russian, Chinese/Mandarin, and other languages planned as well. This provides additional support for English Language Learners. Code Avengers has been used by over 2 million students worldwide.

Promoting Learning

At its core, CS is about solving problems. Digital outcomes, products, and services are developed in order to find innovative solutions to problems and opportunities we face. That's why at Code Avengers, we integrate problem-solving skills into every learning resource.

Divergent thinking is supported in courses that require students to generate original ideas and examine a variety of viable solutions. Students can investigate tasks in a 'non-linear' manner using logic puzzles that explicitly teach decision-making and problem-solving processes on our platform. We aim to create an environment in which students can gain firsthand experience with the concepts of optimization, redundancy, and reliability, as well as their implications for technological system design, development, and maintenance.

Students are empowered to take an active role in choosing the courses that they complete. They have access to a whole course library, with filters that can help them select a pathway. They can complete a course from beginning to end, or jump to a specific lesson. The courses focus on students creating code to demonstrate their understanding of a concept. This means that they are not just watching a video or passively learning but actively engaging in the learning material to create and modify work. Our teacher training emphasizes strategies like "Pair Programming", "Try 3 Then Me", and the importance of choice and space to tinker. Our learning content complements these approaches.

Individualized Learning

Teachers love the self-paced Code Avengers curriculum as it encourages and enables students to be confident, independent learners. Our courses are perfect for students of any experience level and have many tools available for students with diverse learning needs. When designing courses, we use a variety of strategies to support students and ensure learning is accessible.

Students enjoy the Code Avengers platform because of all the features it offers including immediate feedback, helpful videos, pop-up vocabulary, hints, a built-in integrated development environment (IDE), achievement tracking, and other resources. All tasks are self-graded with real-time progress tracking provided on the teacher dashboard and class data can be exported to a spreadsheet at any time.

Teachers can maintain control of class direction by assigning specific modules to class groups or individual students and simultaneously manage class differentiation needs.

Not only can our content be tailored to our students, but our teachers also benefit from the platform's adaptability. Teachers can learn, explore, and upskill alongside their students without having to be experts. Code Avengers makes it easy for teachers with any amount of experience to facilitate effective CS learning in their classrooms. It is a high priority of ours to provide teachers with a suite of resources to support the delivery and monitoring of student learning, including live student tracking, analytics, lesson plans, quizzes, offline activities, and more.

Collaboration

Unplugged activities aimed at helping students think critically and strategically are available to support many of the courses. Students take on specific roles typical in a development cycle to contribute to a project as a team and are taught to manage their time and resources to produce digital artifacts. There are multiple opportunities to collaborate on Code Avengers. Our Pro games and website are shareable and we have JS and HTML/CSS IDEs as practice spaces that allow sharing your code and the ability to download work.

We also provide support and guidance for teachers that includes effective pedagogy and teaching strategies to promote collaboration in Computer Science learning.

Collaboration is fundamental to our internal ethos. We encourage learners and teachers to provide feedback and make suggestions for how to improve our platform, and we actively seek both perspectives when identifying new resources for development.

Real-World Applications

Our courses offer authentic learning opportunities that propose real-world challenges. This allows students to apply their understanding of CS concepts through memorable and compelling applications. Students are expected to use higher-order cognitive processes to analyze problems using a range of skills, such as detail-oriented coding, teamwork, and big-picture, long-term thinking.

Our Junior courses are story-based, and the learning happens within relevant real-world scenarios. At the Pro (senior) level, students write code to build real-world programs and apps that have actual everyday applications.

The impacts of computing are an important topic addressed in many of our courses. Students design and iteratively develop computational artifacts for practical purposes, personal expression, and to address societal issues. They are required to critically evaluate computational artifacts and consider both their beneficial and harmful effects on society. At higher levels, they theorize about the future of the computing innovations that have altered our culture and discuss the legal and ethical implications of these innovations.

Career & Technical Education

Students transition to learning industry-standard programming languages, such as JavaScript and Python, and web languages such as HTML5 and CSS3, or build their theoretical knowledge with our Computer Science and Design courses. These are some of the most in-demand languages in the tech sector, whilst also being very beginner-friendly for both students and teachers. This means learners are given an excellent foundation to move into entry-level/internship-type roles, or on to further study.

Cross-Curricular Connections

Cross-curricular links help learners to make connections with a range of other domains including further STEM topics, social sciences, and the arts. As a company that is already well established in the field of EdTech, Code Avengers is especially well placed to continue to adapt our product to enhance cross-curricular learning. This positions us well to support schools and teachers with the introduction of CS learning as curricula begin to roll out across various states of the USA. We understand that teachers sometimes need to work CS into their classrooms in intelligent ways, so our courses are built to integrate into mainstream learning and connect concepts across the curriculum. When developing new content, cross-curricular connections are a fundamental part of the planning process. Some examples of cross-curricular links found in our existing content are:

- Design theory e.g. color theory.
- History e.g. the history of numbering systems, the evolution of agriculture with AI and agritech.
- Social sciences e.g. the ethics and social impacts of current and future technology.
- Language & literacy e.g. writing and literature-themed activities and writing prompts, procedural writing, constructing texts for specific audiences.
- Science e.g. the physics of light and sound.
- Math e.g. geometry, coordinates, statistics, financial literacy, and more.

Assessment

The Code Avengers learning platform has a number of assessment opportunities available for learners.

Automatic Checking and Validation

Every task on the platform is assessed through automatic checking and validation, to ensure the learner has completed the task properly. This means learners are unable to progress to the next step of their learning without demonstrating an understanding of the step they are currently on. Hints and solutions are available to support learners so that they can self-assess, and practice self-reflection to monitor their own learning. When students use these features, assessment data is generated and displayed as a color-coded progress report in the teacher dashboard view. This means that formative assessment happens naturally as a part of the learning process, and teachers can use this data to identify learners in need of further support without any additional assessment or evaluation work needed on their part.

Quizzes

Our platform also has built-in quizzes that support the recapping and consolidation of information learned in the courses. Lesson plans contain Kahoot quizzes, Google Forms, and other tasks that can also be used to assess student learning for formative or summative purposes.

Projects

Shorter projects that require similar skills are connected to many courses. These have less scaffolding in the instructions in order to support the application of skills learned in the courses. This allows students to try to solve problems more independently and provides teachers with some insight into how well learners have grasped core concepts.

Our step-by-step lesson plans are organized into units that culminate in a project where learners apply the skills they have gained. During this process, learners co-construct and frequently refer to a rubric based on success criteria that they have identified. This gives them the skills they need to be able to feed forward in both self and peer assessment.

Supporting materials

Additional resources such as unplugged activities, workshops, and more, are available to allow for a range of ways to develop and assess learners' understanding of the concepts in the courses.

Standard alignment

Assessment is embedded within all Code Avengers courses. This means that the same coverage, as shown in the curriculum alignment above, applies for both learning content and assessment. The additional resources such as quizzes, projects, and lesson plans that are wrapped around the course content offer opportunities for learning to be assessed in multiple ways.

Performance descriptors

All of our courses come with teaching guides that include detailed breakdowns of the curriculum links, learning outcomes, and task objectives for every lesson and task in the course. These teaching guides also include key vocabulary and definitions so that any misconceptions or misunderstandings can be addressed before teaching and learning are undertaken.

Our growing library of fully guided, step-by-step unit and lesson plans also includes clear curriculum links, learning outcomes, and project options for summatively assessing learning from the unit.

This makes it clear to teachers what learning is being assessed in each lesson, course, and task, and what learners would be expected to be able to do/perform upon completion of the lesson or course.

Assessment Equity

Throughout Code Avengers courses, real-time assessment is carried out using a wide variety of activity and widget types, with all concepts assessed more than once throughout a course, using different learning modes. Courses are designed with diverse learning styles in mind, which means learners get multiple opportunities to apply their learning, in a way that suits them.

Audio is provided throughout many of the courses, particularly those aimed at younger learners, to support those who may struggle with literacy demands, including English Language Learners. Videos are provided in many courses for older learners as well.

Our step-by-step lesson plans offer guidance for differentiated learning in order to support a diverse range of learners, from those who need additional support, to gifted learners. The self-paced nature of Code Avengers, and the ability to assign different courses and modules to various learners, further support teachers to deliver differentiated programs of learning and assessment.

Professional Development

No experience or specific professional development (PD) training is required in order to successfully use Code Avengers to support Computer Science learning in the classroom. However, we do offer a range of resources to support teachers to get the best out of using Code Avengers with their learners.

Our courses are scaffolded to support educators and learners of all experience levels. Teachers can complete courses themselves to upskill or learn alongside students collaboratively. The rich set of resources, with interactive tasks and activities, is ideal for teachers to learn computer science concepts, programming languages, and other applications of real-world skills.

In addition to the full course library, EDU licenses provide access to our interactive Professional Development courses, covering topics such as computer science concepts, collaborative teaching strategies and effective pedagogy for computer science, repositioning the teacher from being the expert to learning alongside their students, fostering perseverance and innovation, and culturally responsive pedagogy. This is valuable learning, given the significant number of teachers who come into teaching Computer Science without a background in the subject, and the importance of being able to engage diverse learners.

Code Avengers is an accredited PD provider, with extensive experience delivering in-person and online professional development programs in numerous settings across the globe. We at Code Avengers will always provide support on our platform to our customers, as an included part of our software as a service. We can also provide custom, personal, and expert professional development to any teacher, school, or district that could benefit from it, in-person or online.

We have consistently and effectively delivered high-quality PD since our inception. Teachers' confidence and time are a top priority, and we can provide additional tools beyond our included support to build confidence and save time.

PD Testimonial (Roz, Maihihi School):

> “This was an excellent piece of PD which was relevant and well facilitated. Thank you so much. The presenters were great. For myself, as a non-confident digital user, they were very considerate and understanding in helping me to work through what was to be taught and the best way to support my students with their digital learning. Thank you for giving me the confidence to be a digital learner and a digital teacher.“

Please contact the Code Avengers team to learn more about our current offerings and schedule. More testimonials are included below.

Research & Testimonials

Research & Case Studies

Queensland Pilot

In late 2019, a trial was conducted with Trinity Bay State High School in Manunda, Queensland Australia. It has a demographic of approximately 35% indigenous learners. Having previously had difficulty engaging their learners with digital technologies, the school piloted the Code Avengers platform to explore whether it would improve student engagement and learning outcomes.

The trial was a resounding success, with student engagement significantly improved, particularly for indigenous girls. The pilot report can be read here, summarized in the quote below with key findings and benefits of using Code Avengers:

> “Differentiation of Digital Technologies course content is now truly possible with the Code Avengers platform. We have a very large Special Needs Program at the school with over 150 students identified with special needs/cognitive disorders/physical issues. Many of these students have worked extremely effectively on the Code Avengers platform. This is an area where their various disabilities are of little consequence. Many of the specialist staff that support them have been extremely surprised at the results that the students have achieved using Code Avengers.”

Chile Pilot

In 2020-2021, a tertiary education provider in Chile ran a pilot program using the Code Avengers platform for their learners. The institution had previously struggled with a failure rate of over 30% for their Foundations of Computer Science course, which covered topics such as the basics of computer programming and web development.

A number of campuses were given access to the platform as well as guidance on which courses learners should complete to support their coursework. A threshold of 2000 points was established, representing the Code Avengers courses best aligned with a particular segment of their coursework. Of the students that met or exceeded the 2000-point threshold, 86% passed the course. This meant a 14% failure rate–

less than half of the previous 30+% failure rate. There was a strong correlation, therefore, between learner engagement with the Code Avengers platform, and pass rates.

Chicago Case Study
In 2012, the Chicago Public Schools recognized a discrepancy: the fastest-growing job sector, Information Technology, was predominantly white and overwhelmingly male. They were determined to shift that balance.

When the Chicago Public Schools began to pilot an “Exploring Computer Science” course several years ago, the staff at Whitney Young High School jumped on board. Their primary goal was to increase diversity in the computer science field - more women, and more people of color. The innovation took hold and before long, the staff at Whitney Young determined that they needed more - more content, more lesson plans, and more assessments. They wanted to take computer science instruction to the next level.

One of the major reasons why CS gained so much traction at Whitney Young was due to them adopting Code Avengers as a tool. Code Avengers helps break down the stereotypes around Computer Science. Students at Whitney Young saw that it was not just the stereotypical math-and-science-minded individuals who can succeed in Computer Science. Many of the students who took the introductory Computer Science course gained confidence and continued on to explore advanced coursework including AP level Computer Science courses. This opened up more opportunities for students to challenge themselves with college-level Computer Science instruction.

Testimonials

The University of Auckland:

> “Our course is a core component of the Bachelor’s Degree of Commerce, with well over 2,000 students enrolled each year. Providing personal training in coding, along with helpful evaluation and feedback for each student in such a course is extremely difficult, if not impossible. Since 2014, codeavengers.com has been supporting our course with its outstanding products, including suitable coding lessons for each individual student and fantastic technical support. The teaching team has been receiving positive feedback from the class about the learning environment on the website, and how it encourages students to further explore different aspects of programming. On the other hand, the student evaluation functionalities on codeavengers.com made it so much easier to mark and monitor students’ progress. Thanks to the team at codeavengers.com, our faculty now has a course that can effectively and efficiently introduce large groups of students to the amazing world of programming.”

David (Head of Department, Tawa College, NZ):

> "**What students like:** We have received positive feedback about the online courses. Students can learn in their own time at their own pace. Nice simple interface, not too much text for instructions with help functions. Also, the level of learning aligns really well with what is expected at years 11, 12, and 13 from NZQA [New Zealand Qualifications Authority].
>
> **What teachers like:** We have appreciated the fact that the courses are at a good level and align with the NZQA Assessments that we set for our students. Examples; Python 1, 2, and 3 all scaffold nicely with students completing the online course for homework during the Term, then as

> teachers we recap the learning on CA in class so students are ready for their final NZQA / NCEA [National Certificate in Educational Achievement] Assessment."

Tatiana (Independent Learner):

> “It is my first time using code avengers. It is a really well-structured page and I enjoy it. I have never seen a code-teaching website that is this good. The rest that I have seen are mediocre at best. Although Khan Academy had some interesting stuff. But I 100% prefer code avengers and I will continue to learn using it.”

Aimee (Teacher, Sage School, MA):

> “Code Avengers is the perfect platform for kids to gain fluency in programming languages. We serve gifted students so the option for kids to independently advance ahead in lessons is crucial."

Whitney Young High School (Chicago, IL)

> “Student work is self-paced, monitored easily through the progress monitoring dashboard, and teachers can use that data to support student learning with one-on-one or small group instruction, pairing students for peer coaching, or offering additional challenges for those who master content quickly.
>
> Code Avengers delivers everything necessary for a school to equip students with the digital technology skills that will start them on the path to be primed to apply for (and land) the computer-science-based jobs of today and tomorrow: a wide range of Computer Science content, pedagogically sound ready-to-use lesson plans, solid teacher support, progress monitoring, engaging and scaffolded lesson plans, and on-target assessments.”

Andrew Mauer-Oats (Whitney Young High School):

> “Code Avengers gives my advanced students the chance to run on ahead; nothing holds them back from learning all that they wish to learn and the platform is loaded with tons of content giving my students the chance to learn HTML, CSS, or Python.”

Technical Specifications

Operating Systems

Code Avengers runs in the browser, so it is compatible with all common operating systems including:

- Windows
- Chrome
- iOS (iPads)
- MacOS

Devices

Code Avengers can be used on a variety of devices, as long as the screen resolution is at least 1024px x 720px, including:

- Chromebooks
- iPads
- Desktop PCs
- Laptops

Coding courses generally require a device with a keyboard, so Chromebooks, laptops, and desktops are recommended for the best experience with HTML/CSS, web development, and programming-type courses.

Web Browsers

Code Avengers works best in Google Chrome, for the most optimized experience. Code Avengers is compatible with these browsers, and others built on the same technology:

- Google Chrome
- Firefox
- Microsoft Edge
- Safari

Learning Management Systems

Code Avengers is fully compatible with both GSuite/Google Classroom and Canvas LMS's. We are compatible with other LMSs of a similar caliber and offer teacher tools that make it easy to share particular courses, lessons, and other URLs within other LMS environments.

Programming Languages Taught

Code Avengers courses cover the following programming languages:

- Blockly
- JavaScript including the PaperScript and jQuery libraries
- Python

In addition, we also cover HTML and CSS.

Hardware Requirements

Apart from a compatible device, there are no specific hardware requirements for Code Avengers content. In the not-too-distant future, we are looking at developing more resources to support hardware-related learning in topics such as computer systems, and electronics-related using equipment such as Microbits. Wherever possible, however, we endeavor to offer virtual alternatives in order to support equity in access as not all teachers and learners have access to hardware such as this.

Additional Information

Teacher Friendliness & Usability

This has largely been addressed throughout this report, but just to reinforce, our materials are developed by teachers, for teachers, and have been used extensively by students. They are continually polished in response to feedback, further improving user-friendliness.

Tools & Materials Required or Recommended

Site licenses are required to access most Code Avengers content and teacher resources. Aside from that, nothing additional is required, as all teaching guides, lesson plans, quizzes, projects, assessment resources, and so on are included with the licenses.

Additional Tools/Materials Available

In addition to the online course content, additional resources as outlined in earlier sections of this report are available. These are all included with the site license, so are not required to be purchased separately.

Free Resources

Code Avengers offers a small number of free courses as tasters. These include our most recent Hour of Code courses, as well as a selection of other courses that is revised from time to time. Free courses also come with teaching guides, and unplugged activities to support further learning.

CODE AVENGERS

QUEENSLAND CURRICULUM (V9)

DIGITAL TECHNOLOGIES ALIGNMENT

QUEENSLAND CURRICULUM (V9)
DIGITAL TECHNOLOGIES ALIGNMENT

The Australian Federal Curriculum Version 9 for Digital Technologies covers learning from Foundation to Year 10. Various Australian states have their own intermediary curriculum documents, particularly Victoria, New South Wales, and Western Australia. The Queensland Curriculum, however, follows the Australian Curriculum as it is written, for Prep to Year 10, and has moved from v8.4 to v9. Each state also has its own senior curriculum and qualifications framework for Years 11-12, as the Federal Curriculum only reaches up to Year 10. Our existing Australian senior secondary alignments can be found here.

This report summarizes the alignment of the Code Avengers platform resources with the curriculum standards used by Queensland. The curriculum content comprises 2 strands, with varying substrands, that span across 6 age bands from Prep-10. **Note:** not every grade has content for all substrands, so this report will group content into the two strands.

Strands & Substrands

1. **Knowledge & Understanding**
    a. Digital Systems
    b. Data Representation

2. **Processes & Production Skills**
    a. Acquiring, managing and analysing data
    b. Investigating and defining
    c. Generating and designing
    d. Producing and implementing
    e. Evaluating
    f. Collaborating and managing
    g. Privacy and security

Grade bands:

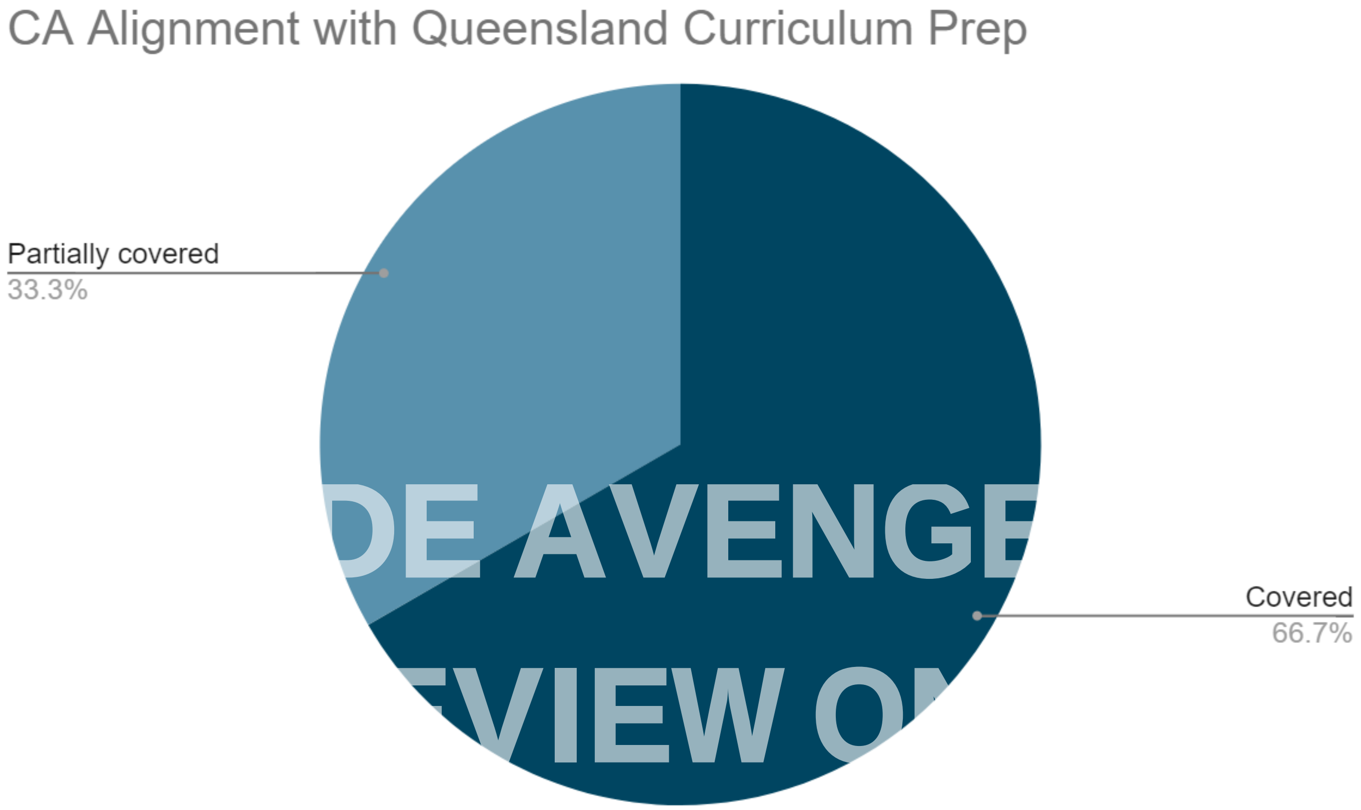
1. Prep
2. Years 1-2
3. Years 3-4
4. Years 5-6
5. Years 7-8
6. Years 9-10

There are 73 content descriptors/curriculum standards from Foundation to Year 10. The tables and charts on the following pages will highlight how well the Code Avengers content currently available aligns with these curriculum standards.

Alignment by Grade Band

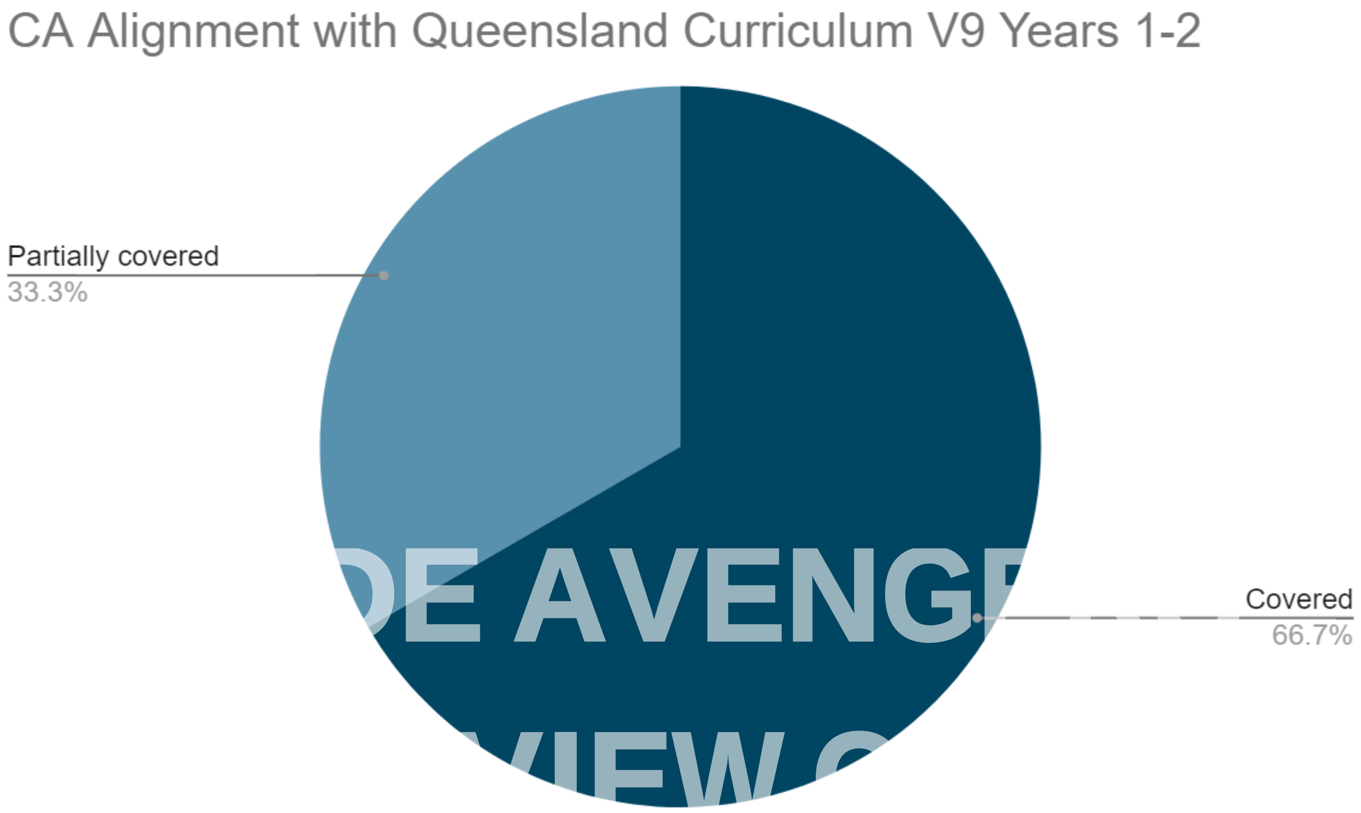
Prep

Topic	Yes	Partial	No
1. Knowledge & Understanding	2	0	0
2. Processes & Production Skills	0	1	0
Totals (out of 3 standards)	**2**	**1**	**0**



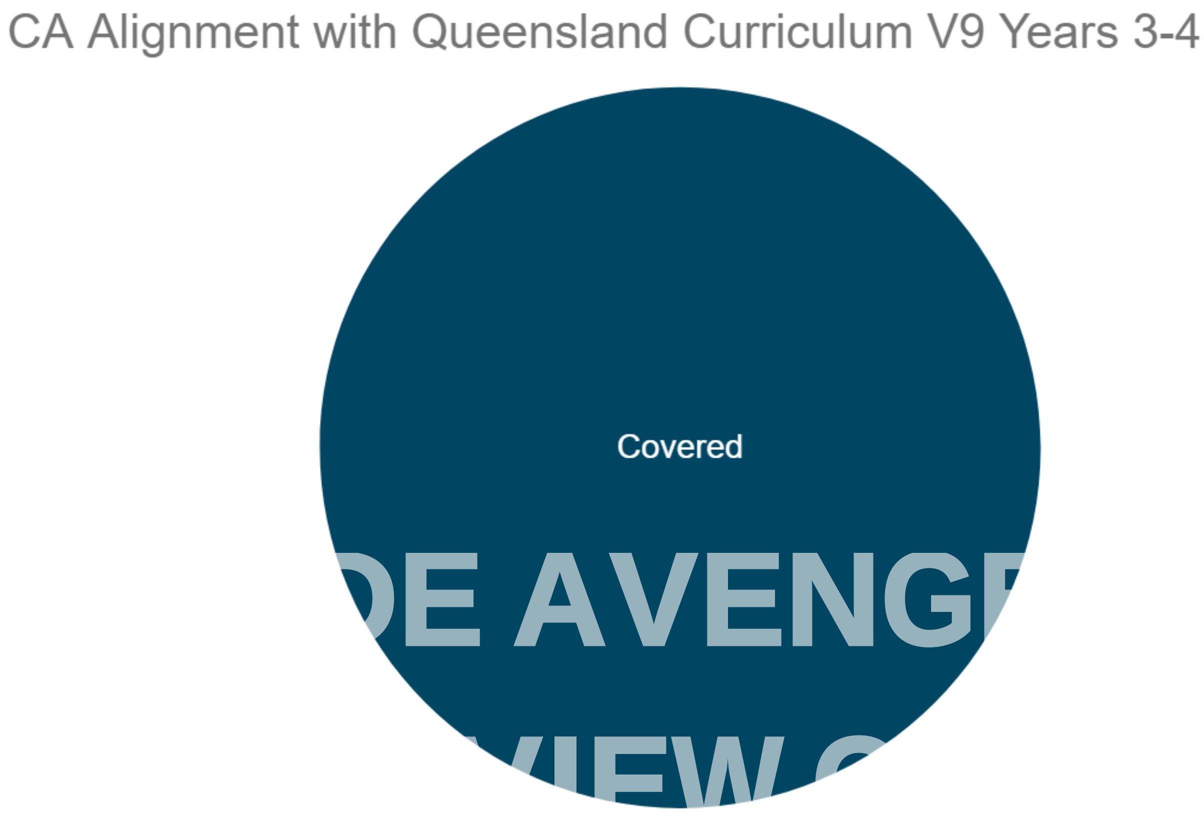
Years 1-2

Topic	Yes	Partial	No
1. Knowledge & Understanding	2	0	0
2. Processes & Production Skills	4	3	0
Totals (out of 9 standards)	**6**	**3**	**0**



Years 3-4

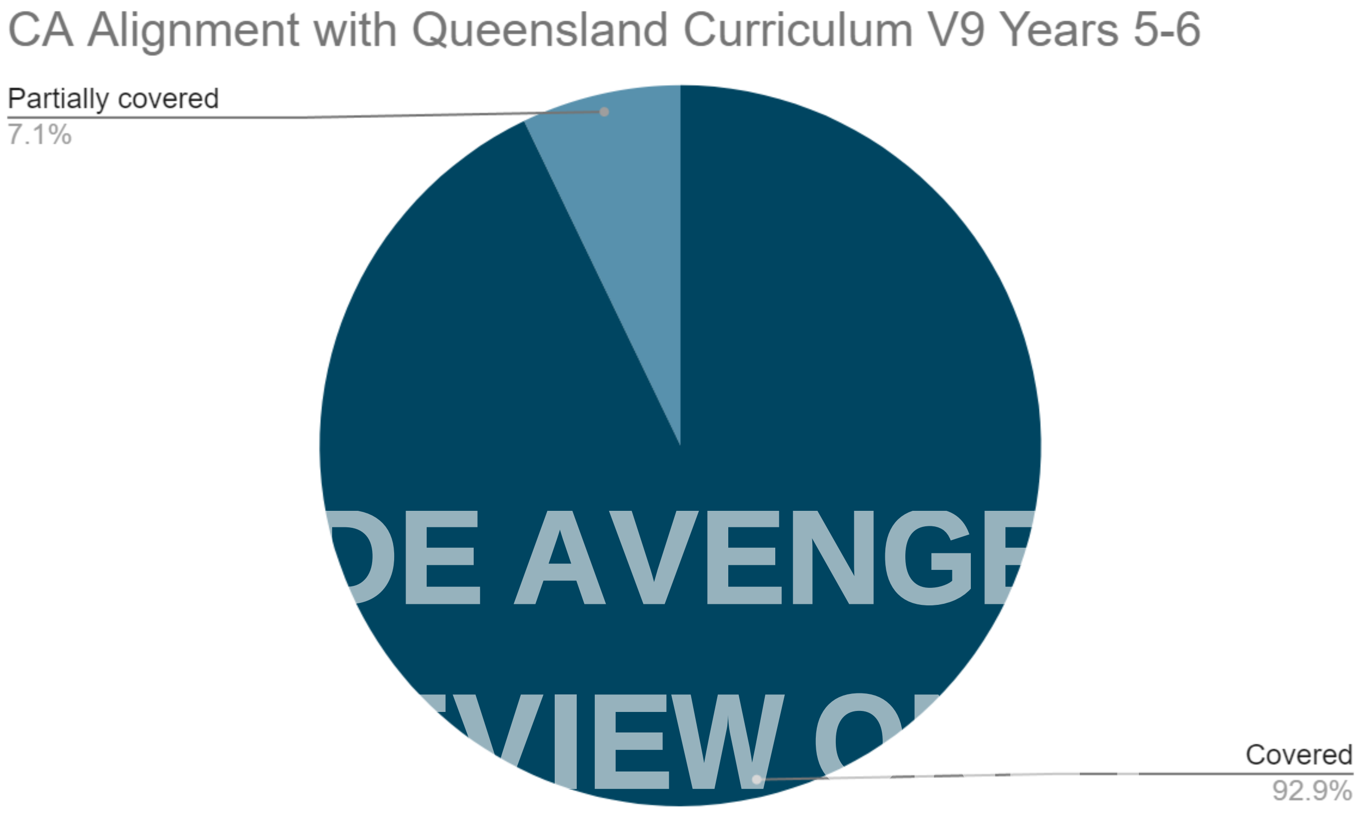
Topic	Yes	Partial	No
1. Knowledge & Understanding	3	0	0
2. Processes & Production Skills	9	0	0
Totals (out of 12 standards)	**12**	**0**	**0**



CA Alignment with Queensland Curriculum V9 Years 3-4

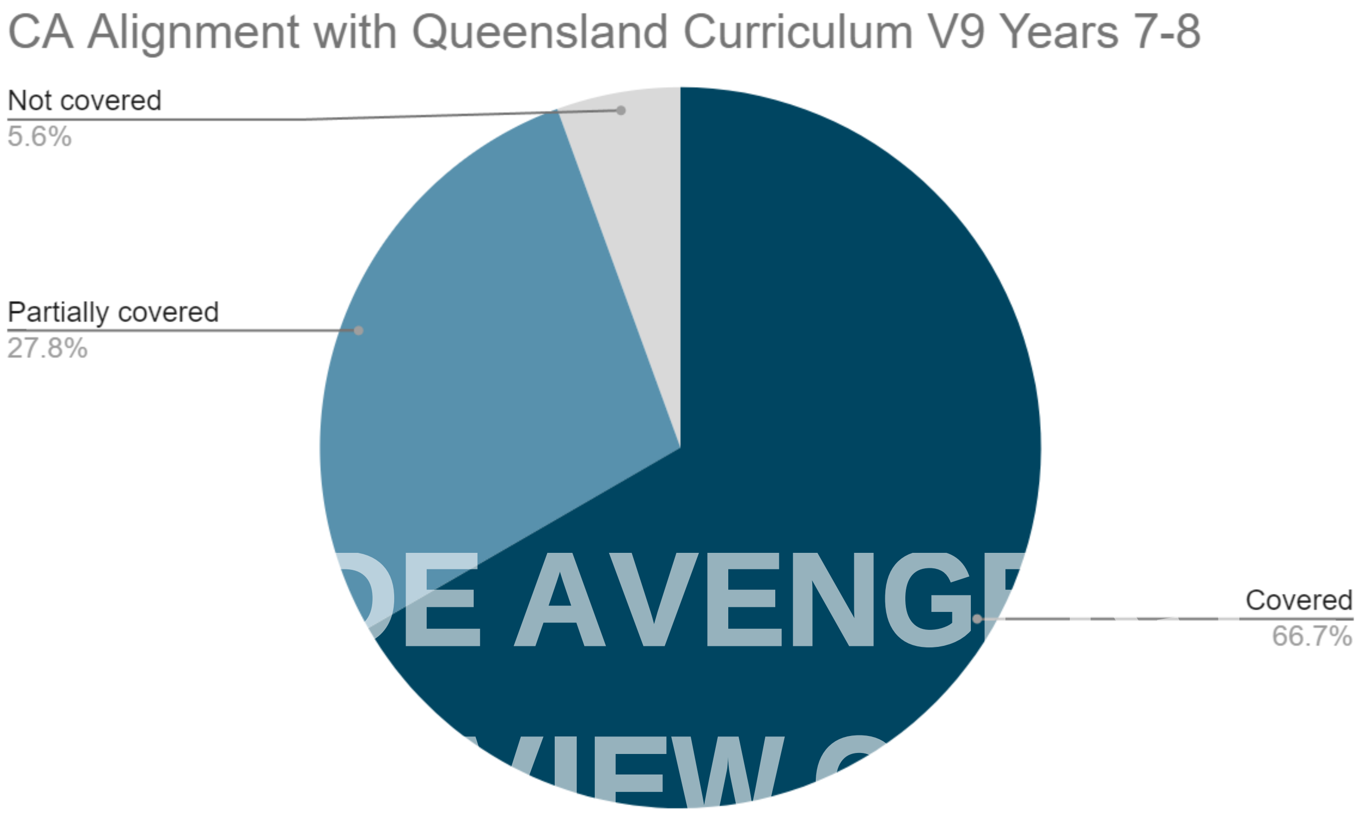
Years 5-6

Topic	Yes	Partial	No
1. Knowledge & Understanding	3	1	0
2. Processes & Production Skills	10	0	0
Totals (out of 14 standards)	**13**	**1**	**0**



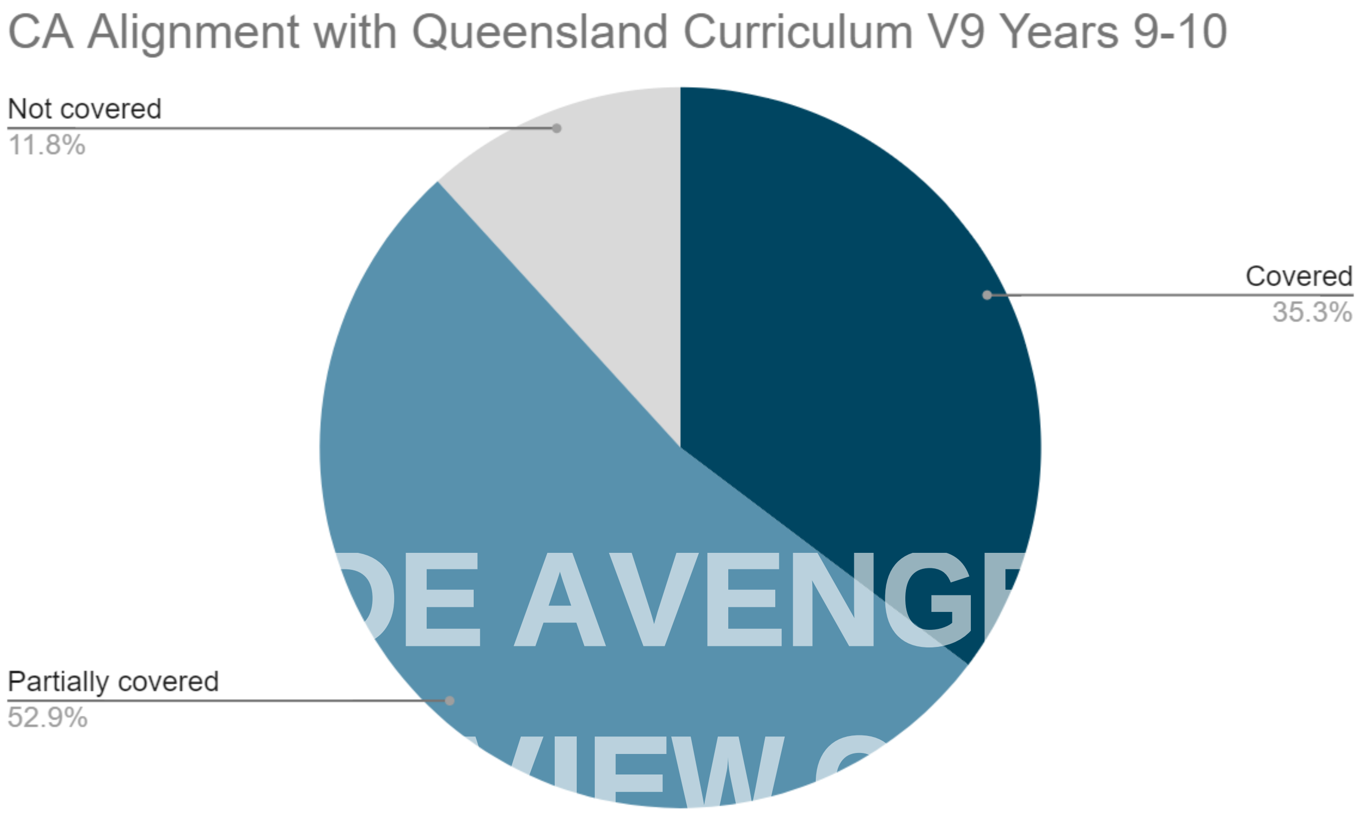
Years 7-8

Topic	Yes	Partial	No
1. Knowledge & Understanding	3	0	1
2. Processes & Production Skills	9	5	0
Totals (out of 18 standards)	**12**	**5**	**1**



Years 9-10

Topic	Yes	Partial	No
1. Knowledge & Understanding	2	0	1
2. Processes & Production Skills	4	9	1
Totals (out of 17 content descriptors)	**6**	**9**	**2**



Appendices follow

Appendix 1: Background & Detail

Code Avengers content covers the Queensland Curriculum V9 for Digital Technologies standards as follows: out of a total of 73 standards, we fully cover 51, partially cover 19, and don't cover 3.

- We fully cover the Years 3-4 band.
- We fully or partially cover all standards Prep-6.
- At Years 7-8, we cover or partially cover 94.4% of standards.
- At years 9-10, we cover or partially cover 88.2% of standards.

A full breakdown of the alignment to the standards is included below.

Key

Color	Explanation
🟩	Code Avengers content covers this standard
*	Code Avengers partially covers this content, and/or it is content that would be covered organically through participation in Digital Technologies learning e.g. logging into school accounts, using devices, and so on.
🟨	Code Avengers content partially covers this standard.
🟥	Code Avengers content does not cover any of this standard at present.

Queensland (V9) DT Alignments

1.	**Prep**	
1.1.	**Knowledge and understanding**	
1.1.1.	Recognise and explore digital systems (hardware and software) for a purpose	🟩
1.1.2.	Represent data as objects, pictures and symbols	🟩
1.2.	**Processes and production skills**	
1.2.1.	Identify some data that is personal and owned by them	🟨
2.	**Years 1-2**	
2.1.	**Knowledge and understanding**	
2.1.1.	Identify and explore digital systems and their components for a purpose	🟩 *
2.1.2.	Represent data as pictures, symbols, numbers and words	🟩
2.2.	**Processes and production skills**	

2.2.1.	Investigate simple problems for known users that can be solved with digital systems	🟩
2.2.2.	Follow and describe algorithms involving a sequence of steps, branching (decisions) and iteration (repetition)	🟩
2.2.3.	Discuss how existing digital systems satisfy identified needs for known users	🟩
2.2.4.	Use the basic features of common digital tools to create, locate and communicate content	🟨
2.2.5.	Use the basic features of common digital tools to share content and collaborate demonstrating agreed behaviours, guided by trusted adults	🟨
2.2.6.	Access their school account with a recorded username and password	*
2.2.7.	Discuss that some websites and apps store their personal data online	🟨
3.	**Years 3-4**	
3.1.	**Knowledge and understanding**	
3.1.1.	Explore and describe a range of digital systems and their peripherals for a variety of purposes	🟩
3.1.2.	Explore transmitting different types of data between digital systems	🟩
3.1.3.	Recognise different types of data and explore how the same data can be represented differently depending on the purpose	🟩
3.2.	**Processes and production skills**	
3.2.1.	Define problems with given design criteria and by co-creating user stories	🟩
3.2.2.	Follow and describe algorithms involving sequencing, comparison operators (branching) and iteration	🟩
3.2.3.	Generate, communicate and compare designs	🟩
3.2.4.	Implement simple algorithms as visual programs involving control structures and input	🟩
3.2.5.	Discuss how existing and student solutions satisfy the design criteria and user stories	🟩
3.2.6.	Use the core features of common digital tools to create, locate and communicate content, following agreed conventions	🟩
3.2.7.	Use the core features of common digital tools to share content, plan tasks, and collaborate, following agreed behaviours, supported by trusted adults	🟩

3.2.8.	Access their school account using a memorised password and explain why it should be easy to remember, but hard for others to guess	🟩
3.2.9.	Identify what personal data is stored and shared in their online accounts and discuss any associated risks	🟩
4.	**Year 5-6**	
4.1.	**Knowledge and understanding**	
4.1.1.	Investigate the main internal components of common digital systems and their function	🟨
4.1.2.	Examine how digital systems form networks to transmit data	🟩
4.1.3.	Explain how digital systems represent all data using numbers	🟩
4.1.4.	Explore how data can be represented by off and on states (zeros and ones in binary)	🟩
4.2.	**Processes and production skills**	
4.2.1.	Define problems with given or co-developed design criteria and by creating user stories	🟩
4.2.2.	Design algorithms involving multiple alternatives (branching) and iteration	🟩
4.2.3.	Design a user interface for a digital system	🟩
4.2.4.	Generate, modify, communicate and evaluate designs	🟩
4.2.5.	Implement algorithms as visual programs involving control structures, variables and input	🟩
4.2.6.	Evaluate existing and student solutions against the design criteria and user stories and their broader community impact	🟩
4.2.7.	Select and use appropriate digital tools effectively to create, locate and communicate content, applying common conventions	🟩
4.2.8.	Select and use appropriate digital tools effectively to share content online, plan tasks and collaborate on projects, demonstrating agreed behaviours	🟩
4.2.9.	Access multiple personal accounts using unique passphrases and explain the risks of password re-use	🟩
4.2.10.	Explain the creation and permanence of their digital footprint and consider privacy when collecting user data	🟩
5.	**Years 7-8**	
5.1.	**Knowledge and understanding**	

5.1.1.	Explain how hardware specifications affect performance and select appropriate hardware for particular tasks and workloads	🟥
5.1.2.	Investigate how data is transmitted and secured in wired and wireless networks including the internet	🟩
5.1.3.	Investigate how digital systems represent text, image and audio data using integers	🟩
5.1.4.	Explain how and why digital systems represent integers in binary	🟩
5.2.	**Processes and production skills**	
5.2.1.	Acquire, store and validate data from a range of sources using software, including spreadsheets and databases	🟩
5.2.2.	Analyse and visualise data using a range of software, including spreadsheets and databases, to draw conclusions and make predictions by identifying trends	🟩
5.2.3.	Model and query the attributes of objects and events using structured data	🟩
5.2.4.	Define and decompose real-world problems with design criteria and by creating user stories	🟨
5.2.5.	Design algorithms involving nested control structures and represent them using flowcharts and pseudocode	🟩
5.2.6.	Trace algorithms to predict output for a given input and to identify errors	🟩
5.2.7.	Design the user experience of a digital system	🟨
5.2.8.	Generate, modify, communicate and evaluate alternative designs	🟩
5.2.9.	Implement, modify and debug programs involving control structures and functions in a general-purpose programming language	🟩
5.2.10.	Evaluate existing and student solutions against the design criteria, user stories and possible future impact	🟨
5.2.11.	Select and use a range of digital tools efficiently, including unfamiliar features, to create, locate and communicate content, consistently applying common conventions	🟨
5.2.12.	Select and use a range of digital tools efficiently and responsibly to share content online, and plan and manage individual and collaborative agile projects	🟨
5.2.13.	Explain how multi-factor authentication protects an account when the password is compromised and identify phishing and other cyber security threats	🟩

5.2.14.	Investigate and manage the digital footprint existing systems and student solutions collect and assess if the data is essential to their purpose	🟩
6.	**Years 9-10**	
6.1.	**Knowledge and understanding**	
6.1.1.	Investigate how hardware and software manage, control and secure access to data in networked digital systems	🟥
6.1.2.	Represent documents online as content (text), structure (markup) and presentation (styling) and explain why such representations are important	🟩
6.1.3.	Investigate simple data compression techniques	🟩
6.2.	**Processes and production skills**	
6.2.1.	Develop techniques to acquire, store and validate data from a range of sources using software, including spreadsheets and databases	🟨
6.2.2.	Analyse and visualise data interactively using a range of software, including spreadsheets and databases, to draw conclusions and make predictions by identifying trends and outliers	🟨
6.2.3.	Model and query entities and their relationships using structured data	🟨
6.2.4.	Define and decompose real-world problems with design criteria and by interviewing stakeholders to create user stories	🟨
6.2.5.	Design algorithms involving logical operators and represent them as flowcharts and pseudocode	🟩
6.2.6.	Validate algorithms and programs by comparing their output against a range of test cases	🟩
6.2.7.	Design and prototype the user experience of a digital system	🟨
6.2.8.	Generate, modify, communicate and critically evaluate alternative designs	🟨
6.2.9.	Implement, modify and debug modular programs, applying selected algorithms and data structures, including in an object-oriented programming language	🟩
6.2.10.	Evaluate existing and student solutions against the design criteria, user stories, possible future impact and opportunities for enterprise	🟩
6.2.11.	Select and use emerging digital tools and advanced features to create and communicate interactive content for a diverse audience	🟨
6.2.12.	Use simple project management tools to plan and manage individual and collaborative agile projects, accounting for risks and responsibilities	🟨

6.2.13.	Develop cyber security threat models, and explore a software, user or software supply chain vulnerability	🟨
6.2.14.	Apply the Australian Privacy Principles to critique and manage the digital footprint that existing systems and student solutions collect	🟥

Appendix 2: Code Avengers Deep Dive

Why Code Avengers?

Background

Code Avengers is the commercial realization of the results of a prolonged and extensive Ph.D. research project that identified 29 core elements required to successfully teach coding. It is a comprehensive online education platform (https://codeavengers.com) that effectively caters to educational institutes and individual learners. Our platform provides interactive, gamified courses that teach computer programming, web development, computational thinking, digital skills, design, and more, through an engaging and effective, “learn-by-doing” approach.

Instruction & Support

Our language vocabulary is tailored to support ESOL learners, including strategies like having both written and spoken instructions and limiting non-jargon vocabulary to 2 thousand most common English words. As a web-based platform, screen reading software makes our courses accessible and all tasks work similarly and have a predictable layout, with audio available for most courses to help reduce literacy barriers.

Our Professional Development courses help teachers develop inclusive teaching practices and strategies to support the learning and achievement of diverse learners with specific needs using digital technologies. Through our support and contact functionality, educators have access to a team of PD facilitators and content developers who support them in a variety of ways, such as assisting them in curriculum development, using our platform to plan individualized instruction, and offering hybrid/ inclusive learning opportunities.

Student Engagement

Our course developers are qualified, experienced teachers, who are experts in both the content and pedagogy of Computer Science (CS). They have been instrumental in creating quality content that separates us from other platforms by effectively engaging learners from a diverse range of backgrounds. Our scaffolded learning methods build students’ skills gradually which means they are continuously challenged and engaged in the learning process.

At Code Avengers, our whole business is focused on providing innovative, future-focused CS education that empowers all learners to develop the digital skills they need to be the creators and innovators of the future. Our course developers have expertise in inclusiveness and diversity, and Code Avengers courses are therefore designed with culturally responsive learning in mind. Because our team is made up of a diverse range of people with different cultures, genders, and experiences, we are able to provide diverse learners with relatable learning contexts and meaningful activities.

We ensure underserved populations can see themselves reflected within the industry and learning environment. Our courses have a diverse cast of characters from varied cultures, countries, and families. At Code Avengers, we support and encourage people from all walks of life to get excited about being creators and innovators in the digital space. The majority of our content is available in English and Spanish, with a growing library of content in German, French, Japanese, Maori, Dutch, Portuguese, Russian, Chinese/Mandarin, and other languages planned as well. This provides additional support for English Language Learners. Code Avengers has been used by over 2 million students worldwide.

Promoting Learning

At its core, CS is about solving problems. Digital outcomes, products, and services are developed in order to find innovative solutions to problems and opportunities we face. That's why at Code Avengers, we integrate problem-solving skills into every learning resource.

Divergent thinking is supported in courses that require students to generate original ideas and examine a variety of viable solutions. Students can investigate tasks in a 'non-linear' manner using logic puzzles that explicitly teach decision-making and problem-solving processes on our platform. We aim to create an environment in which students can gain firsthand experience with the concepts of optimization, redundancy, and reliability, as well as their implications for technological system design, development, and maintenance.

Students are empowered to take an active role in choosing the courses that they complete. They have access to a whole course library, with filters that can help them select a pathway. They can complete a course from beginning to end, or jump to a specific lesson. The courses focus on students creating code to demonstrate their understanding of a concept. This means that they are not just watching a video or passively learning but actively engaging in the learning material to create and modify work. Our teacher training emphasizes strategies like "Pair Programming", "Try 3 Then Me", and the importance of choice and space to tinker. Our learning content complements these approaches.

Individualized Learning

Teachers love the self-paced Code Avengers curriculum as it encourages and enables students to be confident, independent learners. Our courses are perfect for students of any experience level and have many tools available for students with diverse learning needs. When designing courses, we use a variety of strategies to support students and ensure learning is accessible.

Students enjoy the Code Avengers platform because of all the features it offers including immediate feedback, helpful videos, pop-up vocabulary, hints, a built-in integrated development environment (IDE), achievement tracking, and other resources. All tasks are self-graded with real-time progress tracking provided on the teacher dashboard and class data can be exported to a spreadsheet at any time. Teachers can maintain control of class direction by assigning specific modules to class groups or individual students and simultaneously manage class differentiation needs.

Not only can our content be tailored to our students, but our teachers also benefit from the platform's adaptability. Teachers can learn, explore, and upskill alongside their students without having to be experts. Code Avengers makes it easy for teachers with any amount of experience to facilitate effective CS learning in their classrooms. It is a high priority of ours to provide teachers with a suite of resources to support the delivery and monitoring of student learning, including live student tracking, analytics, lesson plans, quizzes, offline activities, and more.

Collaboration

Unplugged activities aimed at helping students think critically and strategically are available to support many of the courses. Students take on specific roles typical in a development cycle to contribute to a project as a team and are taught to manage their time and resources to produce digital artifacts. There are multiple opportunities to collaborate on Code Avengers. Our Pro games and website are shareable and we have JS and HTML/CSS IDEs as practice spaces that allow sharing your code and the ability to download work.

We also provide support and guidance for teachers that includes effective pedagogy and teaching strategies to promote collaboration in Computer Science learning.

Collaboration is fundamental to our internal ethos. We encourage learners and teachers to provide feedback and make suggestions for how to improve our platform, and we actively seek both perspectives when identifying new resources for development.

Real-World Applications

Our courses offer authentic learning opportunities that propose real-world challenges. This allows students to apply their understanding of CS concepts through memorable and compelling applications. Students are expected to use higher-order cognitive processes to analyze problems using a range of skills, such as detail-oriented coding, teamwork, and big-picture, long-term thinking.

Our Junior courses are story-based, and the learning happens within relevant real-world scenarios. At the Pro (senior) level, students write code to build real-world programs and apps that have actual everyday applications.

The impacts of computing are an important topic addressed in many of our courses. Students design and iteratively develop computational artifacts for practical purposes, personal expression, and to address societal issues. They are required to critically evaluate computational artifacts and consider both their beneficial and harmful effects on society. At higher levels, they theorize about the future of the computing innovations that have altered our culture and discuss the legal and ethical implications of these innovations.

Career & Technical Education

Students transition to learning industry-standard programming languages, such as JavaScript and Python, and web languages such as HTML5 and CSS3, or build their theoretical knowledge with our Computer Science and Design courses. These are some of the most in-demand languages in the tech sector, whilst also being very beginner-friendly for both students and teachers. This means learners are given an excellent foundation to move into entry-level/internship-type roles, or on to further study.

Cross-Curricular Connections

Cross-curricular links help learners to make connections with a range of other domains including further STEM topics, social sciences, and the arts. As a company that is already well established in the field of EdTech, Code Avengers is especially well placed to continue to adapt our product to enhance cross-curricular learning. This positions us well to support schools and teachers with the introduction of CS learning as curricula begin to roll out across various states of the USA. We understand that teachers sometimes need to work CS into their classrooms in intelligent ways, so our courses are built to integrate into mainstream learning and connect concepts across the curriculum. When developing new content, cross-curricular connections are a fundamental part of the planning process. Some examples of cross-curricular links found in our existing content are:

- Design theory e.g. color theory.
- History e.g. the history of numbering systems, the evolution of agriculture with AI and agritech.
- Social sciences e.g. the ethics and social impacts of current and future technology.
- Language & literacy e.g. writing and literature-themed activities and writing prompts, procedural writing, constructing texts for specific audiences.
- Science e.g. the physics of light and sound.
- Math e.g. geometry, coordinates, statistics, financial literacy, and more.

Assessment

The Code Avengers learning platform has a number of assessment opportunities available for learners.

Automatic Checking and Validation
Every task on the platform is assessed through automatic checking and validation, to ensure the learner has completed the task properly. This means learners are unable to progress to the next step of their learning without demonstrating an understanding of the step they are currently on. Hints and solutions are available to support learners so that they can self-assess, and practice self-reflection to monitor their own learning. When students use these features, assessment data is generated and displayed as a color-coded progress report in the teacher dashboard view. This means that formative assessment happens naturally as a part of the learning process, and teachers can use this data to identify learners in need of further support without any additional assessment or evaluation work needed on their part.

Quizzes
Our platform also has built-in quizzes that support the recapping and consolidation of information learned in the courses. Lesson plans contain Kahoot! quizzes, Google Forms, and other tasks that can also be used to assess student learning for formative or summative purposes.

Projects
Shorter projects that require similar skills are connected to many courses. These have less scaffolding in the instructions in order to support the application of skills learned in the courses. This allows students to try to solve problems more independently and provides teachers with some insight into how well learners have grasped core concepts.

Our step-by-step lesson plans are organized into units that culminate in a project where learners apply the skills they have gained. During this process, learners co-construct and frequently refer to a rubric based on success criteria that they have identified. This gives them the skills they need to be able to feed forward in both self and peer assessment.

Supporting materials
Additional resources such as unplugged activities, workshops, and more, are available to allow for a range of ways to develop and assess learners' understanding of the concepts in the courses.

Standard alignment

Assessment is embedded within all Code Avengers courses. This means that the same coverage, as shown in the curriculum alignment above, applies for both learning content and assessment. The additional resources such as quizzes, projects, and lesson plans that are wrapped around the course content offer opportunities for learning to be assessed in multiple ways.

Performance descriptors

All of our courses come with teaching guides that include detailed breakdowns of the curriculum links, learning outcomes, and task objectives for every lesson and task in the course. These teaching guides also include key vocabulary and definitions so that any misconceptions or misunderstandings can be addressed before teaching and learning are undertaken.

Our growing library of fully guided, step-by-step unit and lesson plans also includes clear curriculum links, learning outcomes, and project options for summatively assessing learning from the unit.

This makes it clear to teachers what learning is being assessed in each lesson, course, and task, and what learners would be expected to be able to do/perform upon completion of the lesson or course.

Assessment Equity

Throughout Code Avengers courses, real-time assessment is carried out using a wide variety of activity and widget types, with all concepts assessed more than once throughout a course, using different learning modes. Courses are designed with diverse learning styles in mind, which means learners get multiple opportunities to apply their learning, in a way that suits them.

Audio is provided throughout many of the courses, particularly those aimed at younger learners, to support those who may struggle with literacy demands, including English Language Learners. Videos are provided in many courses for older learners as well.

Our step-by-step lesson plans offer guidance for differentiated learning in order to support a diverse range of learners, from those who need additional support, to gifted learners. The self-paced nature of Code Avengers, and the ability to assign different courses and modules to various learners, further support teachers to deliver differentiated programs of learning and assessment.

Professional Development

No experience or specific professional development (PD) training is required in order to successfully use Code Avengers to support Computer Science learning in the classroom. However, we do offer a range of resources to support teachers to get the best out of using Code Avengers with their learners.

Our courses are scaffolded to support educators and learners of all experience levels. Teachers can complete courses themselves to upskill or learn alongside students collaboratively. The rich set of resources, with interactive tasks and activities, is ideal for teachers to learn computer science concepts, programming languages, and other applications of real-world skills.

In addition to the full course library, EDU licenses provide access to our interactive Professional Development courses, covering topics such as computer science concepts, collaborative teaching strategies and effective pedagogy for computer science, repositioning the teacher from being the expert to learning alongside their students, fostering perseverance and innovation, and culturally responsive pedagogy. This is valuable learning, given the significant number of teachers who come into teaching Computer Science without a background in the subject, and the importance of being able to engage diverse learners.

Code Avengers is an accredited PD provider, with extensive experience delivering in-person and online professional development programs in numerous settings across the globe. We at Code Avengers will always provide support on our platform to our customers, as an included part of our software as a service. We can also provide custom, personal, and expert professional development to any teacher, school, or district that could benefit from it, in-person or online.

We have consistently and effectively delivered high-quality PD since our inception. Teachers' confidence and time are a top priority, and we can provide additional tools beyond our included support to build confidence and save time.

PD Testimonial (Roz, Maihihi School):

> “This was an excellent piece of PD which was relevant and well facilitated. Thank you so much. The presenters were great. For myself, as a non-confident digital user, they were very considerate and understanding in helping me to work through what was to be taught and the best way to support my students with their digital learning. Thank you for giving me the confidence to be a digital learner and a digital teacher.“

Please contact the Code Avengers team to learn more about our current offerings and schedule. More testimonials are included below.

Research & Testimonials

Research & Case Studies

Queensland Pilot
In late 2019, a trial was conducted with Trinity Bay State High School in Manunda, Queensland Australia. It has a demographic of approximately 35% indigenous learners. Having previously had difficulty engaging their learners with digital technologies, the school piloted the Code Avengers platform to explore whether it would improve student engagement and learning outcomes.

The trial was a resounding success, with student engagement significantly improved, particularly for indigenous girls. The pilot report can be read here, summarized in the quote below with key findings and benefits of using Code Avengers:

> “Differentiation of Digital Technologies course content is now truly possible with the Code Avengers platform. We have a very large Special Needs Program at the school with over 150 students identified with special needs/cognitive disorders/physical issues. Many of these students have worked extremely effectively on the Code Avengers platform. This is an area where their various disabilities are of little consequence. Many of the specialist staff that support them have been extremely surprised at the results that the students have achieved using Code Avengers.”

Chile Pilot
In 2020-2021, a tertiary education provider in Chile ran a pilot program using the Code Avengers platform for their learners. The institution had previously struggled with a failure rate of over 30% for their Foundations of Computer Science course, which covered topics such as the basics of computer programming and web development.

A number of campuses were given access to the platform as well as guidance on which courses learners should complete to support their coursework. A threshold of 2000 points was established, representing the Code Avengers courses best aligned with a particular segment of their coursework. Of the students that met or exceeded the 2000-point threshold, 86% passed the course. This meant a 14% failure rate–less than half of the previous 30+% failure rate. There was a strong correlation, therefore, between learner engagement with the Code Avengers platform, and pass rates.

Chicago Case Study
In 2012, the Chicago Public Schools recognized a discrepancy: the fastest-growing job sector, Information Technology, was predominantly white and overwhelmingly male. They were determined to shift that balance.

When the Chicago Public Schools began to pilot an “Exploring Computer Science” course several years ago, the staff at Whitney Young High School jumped on board. Their primary goal was to increase diversity in the computer science field - more women, and more people of color. The innovation took hold and before long, the staff at Whitney Young determined that they needed more - more content, more lesson plans, and more assessments. They wanted to take computer science instruction to the next level.

One of the major reasons why CS gained so much traction at Whitney Young was due to them adopting Code Avengers as a tool. Code Avengers helps break down the stereotypes around Computer Science. Students at Whitney Young saw that it was not just the stereotypical math-and-science-minded individuals who can succeed in Computer Science. Many of the students who took the introductory Computer Science course gained confidence and continued on to explore advanced coursework including AP level Computer Science courses. This opened up more opportunities for students to challenge themselves with college-level Computer Science instruction.

Testimonials

The University of Auckland:

> “Our course is a core component of the Bachelor’s Degree of Commerce, with well over 2,000 students enrolled each year. Providing personal training in coding, along with helpful evaluation and feedback for each student in such a course is extremely difficult, if not impossible. Since 2014, codeavengers.com has been supporting our course with its outstanding products, including suitable coding lessons for each individual student and fantastic technical support. The teaching team has been receiving positive feedback from the class about the learning environment on the website, and how it encourages students to further explore different aspects of programming. On the other hand, the student evaluation functionalities on codeavengers.com made it so much easier to mark and monitor students’ progress. Thanks to the team at codeavengers.com, our faculty now has a course that can effectively and efficiently introduce large groups of students to the amazing world of programming.”

David (Head of Department, Tawa College, NZ):

> "**What students like:** We have received positive feedback about the online courses. Students can learn in their own time at their own pace. Nice simple interface, not too much text for instructions with help functions. Also, the level of learning aligns really well with what is expected at years 11, 12, and 13 from NZQA [New Zealand Qualifications Authority].
>
> **What teachers like:** We have appreciated the fact that the courses are at a good level and align with the NZQA Assessments that we set for our students. Examples; Python 1, 2, and 3 all scaffold nicely with students completing the online course for homework during the Term, then as teachers we recap the learning on CA in class so students are ready for their final NZQA / NCEA [National Certificate in Educational Achievement] Assessment."

Tatiana (Independent Learner):

> “It is my first time using code avengers. It is a really well-structured page and I enjoy it. I have never seen a code-teaching website that is this good. The rest that I have seen are mediocre at best. Although Khan Academy had some interesting stuff. But I 100% prefer code avengers and I will continue to learn using it.”

Aimee (Teacher, Sage School, MA):

> “Code Avengers is the perfect platform for kids to gain fluency in programming languages. We serve gifted students so the option for kids to independently advance ahead in lessons is crucial."

Whitney Young High School (Chicago, IL)

> “Student work is self-paced, monitored easily through the progress monitoring dashboard, and teachers can use that data to support student learning with one-on-one or small group instruction, pairing students for peer coaching, or offering additional challenges for those who master content quickly.
>
> Code Avengers delivers everything necessary for a school to equip students with the digital technology skills that will start them on the path to be primed to apply for (and land) the computer-science-based jobs of today and tomorrow: a wide range of Computer Science content, pedagogically sound ready-to-use lesson plans, solid teacher support, progress monitoring, engaging and scaffolded lesson plans, and on-target assessments.”

Andrew Mauer-Oats (Whitney Young High School):

> “Code Avengers gives my advanced students the chance to run on ahead; nothing holds them back from learning all that they wish to learn and the platform is loaded with tons of content giving my students the chance to learn HTML, CSS, or Python.”

Technical Specifications

Operating Systems

Code Avengers runs in the browser, so it is compatible with all common operating systems including:
- Windows
- Chrome
- iOS (iPads)
- MacOS

Devices

Code Avengers can be used on a variety of devices, as long as the screen resolution is at least 1024px x 720px, including:
- Chromebooks
- iPads
- Desktop PCs
- Laptops

Coding courses generally require a device with a keyboard, so Chromebooks, laptops, and desktops are recommended for the best experience with HTML/CSS, web development, and programming-type courses.

Web Browsers

Code Avengers works best in Google Chrome, for the most optimized experience. Code Avengers is compatible with these browsers, and others built on the same technology:

- Google Chrome
- Firefox
- Microsoft Edge
- Safari

Learning Management Systems

Code Avengers is fully compatible with both GSuite/Google Classroom and Canvas LMS's. We are compatible with other LMSs of a similar caliber and offer teacher tools that make it easy to share particular courses, lessons, and other URLs within other LMS environments.

Programming Languages Taught

Code Avengers courses cover the following programming languages:

- Blockly
- JavaScript including the PaperScript and jQuery libraries
- Python

In addition, we also cover HTML and CSS.

Hardware Requirements

Apart from a compatible device, there are no specific hardware requirements for Code Avengers content. In the not-too-distant future, we are looking at developing more resources to support hardware-related learning in topics such as computer systems, and electronics-related using equipment such as Microbits. Wherever possible, however, we endeavor to offer virtual alternatives in order to support equity in access as not all teachers and learners have access to hardware such as this.

Additional Information

Teacher Friendliness & Usability

This has largely been addressed throughout this report, but just to reinforce, our materials are developed by teachers, for teachers, and have been used extensively by students. They are continually polished in response to feedback, further improving user-friendliness.

Tools & Materials Required or Recommended

Site licenses are required to access most Code Avengers content and teacher resources. Aside from that, nothing additional is required, as all teaching guides, lesson plans, quizzes, projects, assessment resources, and so on are included with the licenses.

Additional Tools/Materials Available

In addition to the online course content, additional resources as outlined in earlier sections of this report are available. These are all included with the site license, so are not required to be purchased separately.

Free Resources

Code Avengers offers a small number of free courses as tasters. These include our most recent Hour of Code courses, as well as a selection of other courses that is revised from time to time. Free courses also come with teaching guides, and unplugged activities to support further learning.

CODE AVENGERS

TEXAS ESSENTIAL KNOWLEDGE & SKILLS

CODE AVENGERS STEM TEKS CROSSWALK ALIGNMENT

TEXAS ESSENTIAL KNOWLEDGE & SKILLS

CODE AVENGERS STEM TEKS CROSSWALK ALIGNMENT

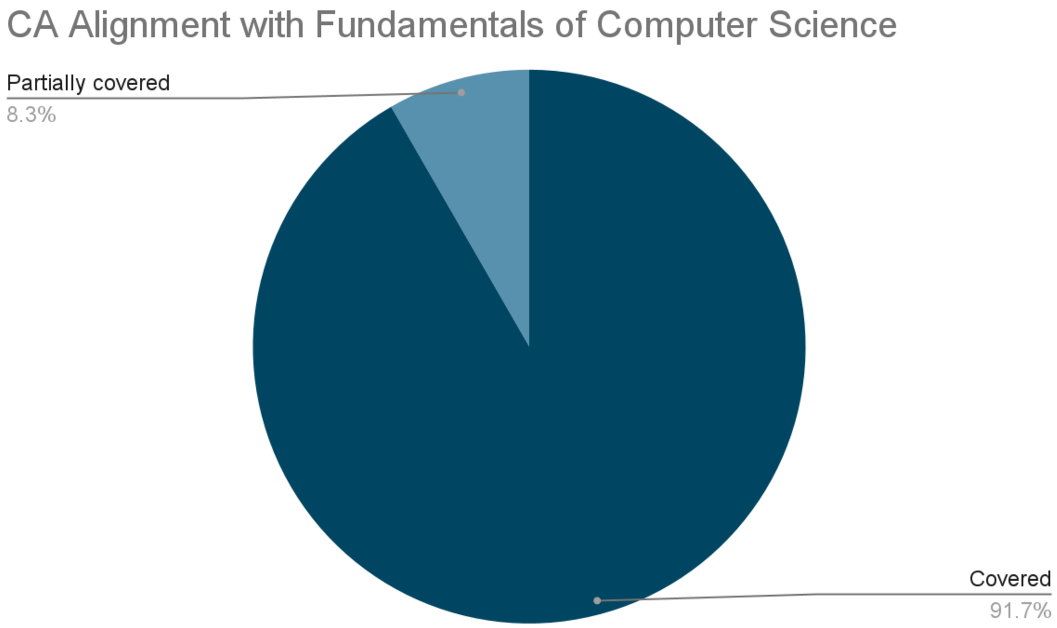
The following 4 courses are completed across grades 9-12. The technology applications curriculum has strands based on performance indicators developed by the International Society for Technology in Education (ISTE):

1. Employability
2. Creativity and innovation
3. Communication and collaboration
4. Data literacy and management
5. Critical thinking, problem solving, and decision making
6. Testing and documentation
7. Technology operations, systems, and concepts
8. Digital citizenship
9. Programming style and presentation

Alignment by Course

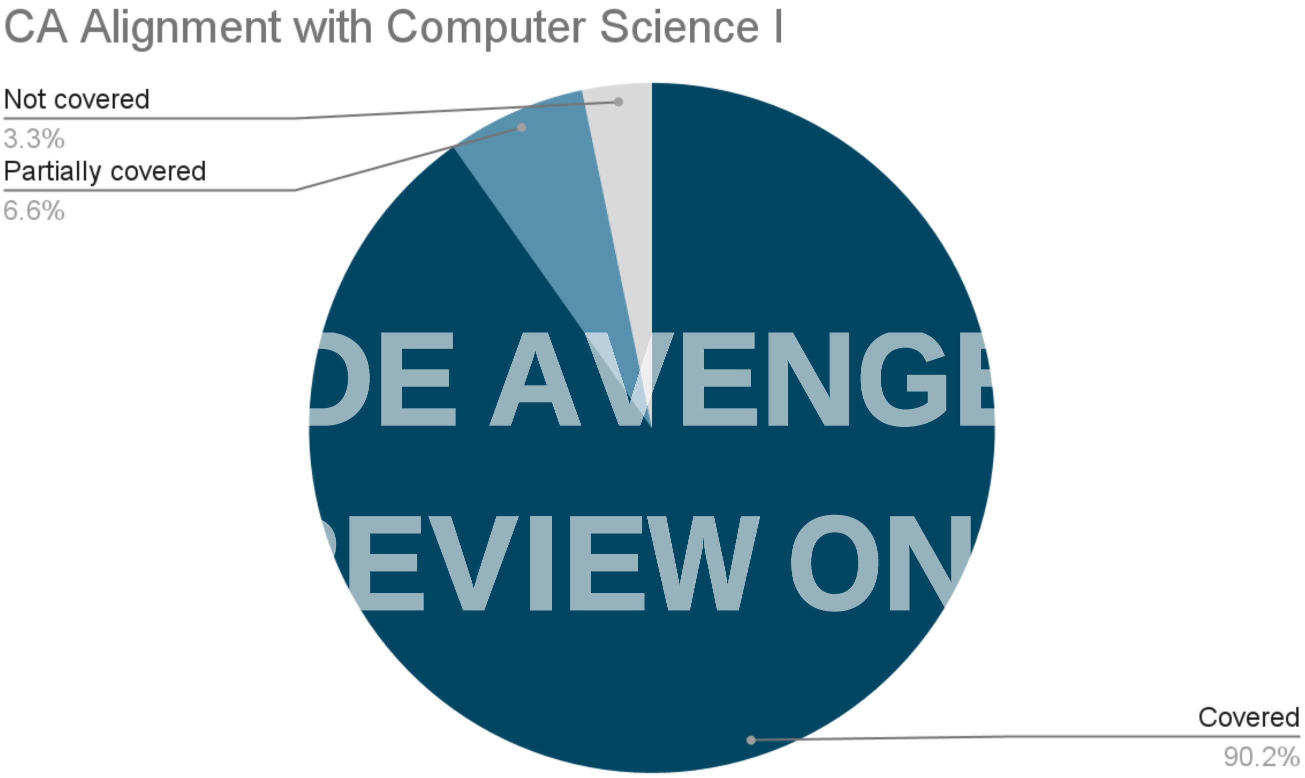
Fundamentals of Computer Science

Topic	Yes	Partial	No
1. Employability	4	1	0
2. Creativity and innovation	4	0	0
3. Communication and collaboration	3	0	0
4. Critical thinking, problem solving, and decision making	13	0	0
5. Digital citizenship	4	2	0
6. Technology operations and concepts	5	0	0
Totals (out of 36 performance indicators)	**33**	**3**	**0**



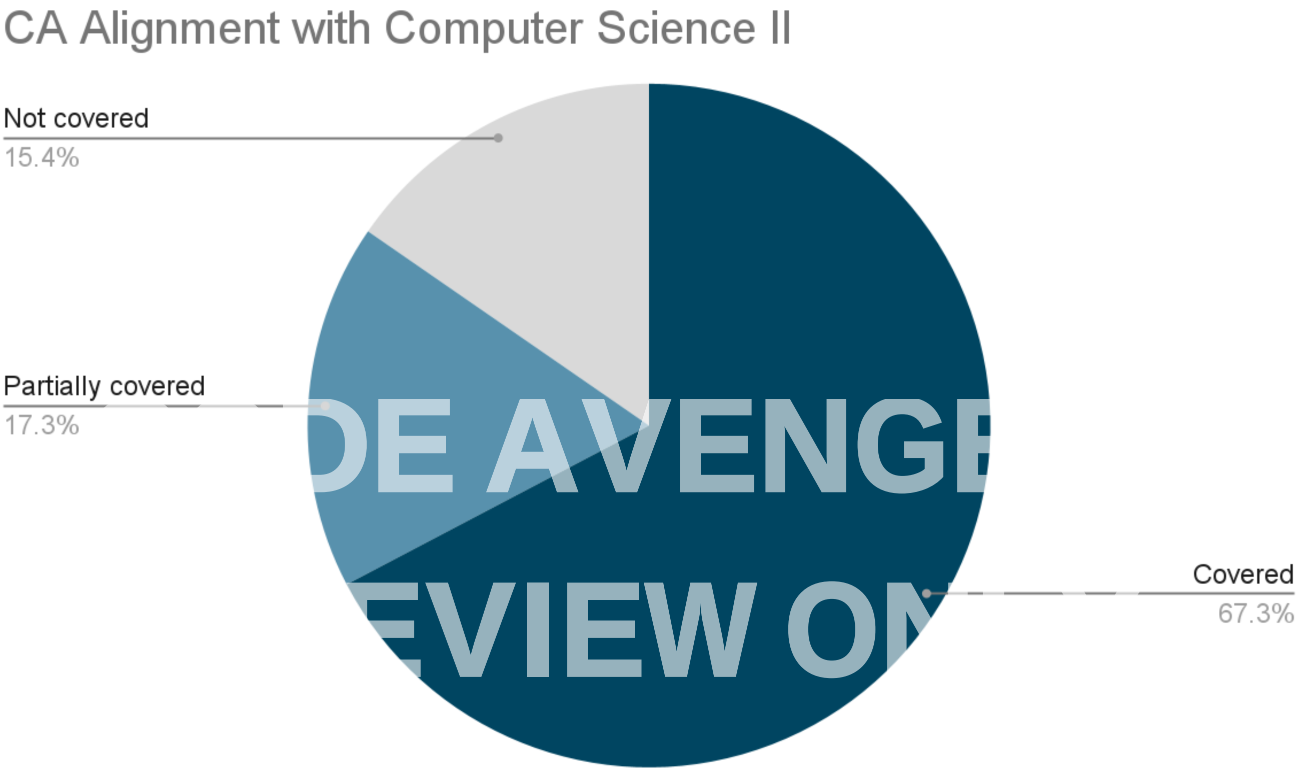
Computer Science I

Topic	Yes	Partial	No
1. Employability	9	0	0
2. Communication and collaboration	2	0	0
3. Programming style and presentation	4	0	0
4. Critical thinking, problem solving, and decision making	20	2	0
5. Digital citizenship	4	1	0
6. Technology operations and concepts	16	1	2
Totals (out of 61 performance indicators)	**55**	**4**	**2**



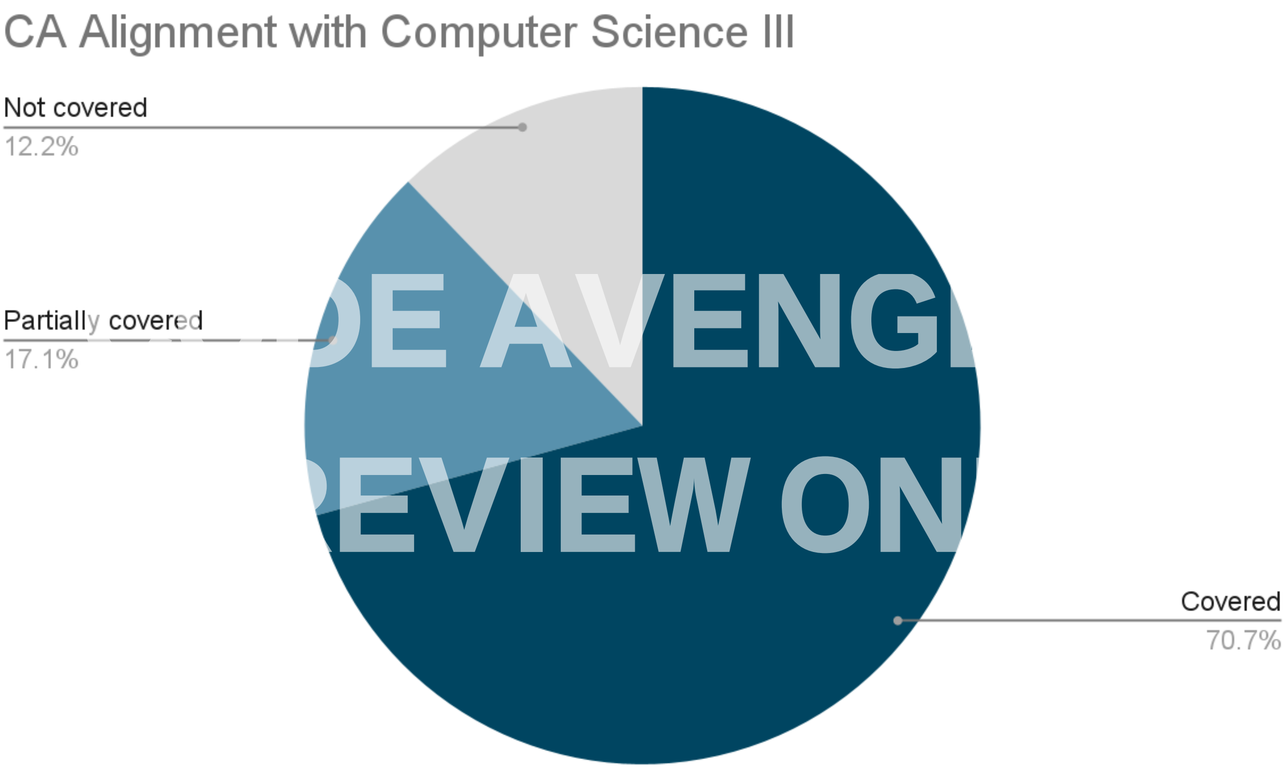
Computer Science II

Topic	Yes	Partial	No
1. Employability	9	0	0
2. Creativity and innovation	4	1	2
3. Communication and collaboration	5	1	0
4. Data literacy and management	8	0	0
5. Critical thinking, problem solving, and decision making	9	7	6
Totals (out of 52 performance indicators)	**35**	**9**	**8**



Computer Science III

Topic	Yes	Partial	No
1. Employability	9	0	0
2. Creativity and innovation	3	1	1
3. Communication and collaboration	1	1	0
4. Data literacy and management	8	0	0
5. Critical thinking, problem solving, and decision making	2	1	3
6. Testing and documentation	4	0	0
7. Practical application of technology	2	4	1
Totals (out of 41 performance indicators)	**29**	**7**	**5**



Why Code Avengers?

Background

Code Avengers is the commercial realization of the results of a prolonged and extensive Ph.D. research project that identified 29 core elements required to successfully teach coding. It is a comprehensive online education platform (https://codeavengers.com) that effectively caters to educational institutes and individual learners. Our platform provides interactive, gamified courses that teach computer programming, web development, computational thinking, digital skills, design, and more, through a fun and effective learn-by-doing approach.

Student Engagement

Our course developers are qualified, experienced teachers, who are experts in both the content and pedagogy of Computer Science (CS). They have been instrumental in creating quality content that separates us from other platforms by effectively engaging learners from a diverse range of backgrounds. Our scaffolded learning methods build students' skills gradually which means they are continuously challenged and engaged in the learning process.

At Code Avengers, our whole business is focused on providing innovative, future-focused CS education that empowers all learners to develop the digital skills they need to be the creators and innovators of the future. Our course developers have expertise in inclusiveness and diversity, and Code Avengers courses are therefore designed with culturally responsive learning in mind. Because our team is made up of a diverse range of people with different cultures, genders, and experiences, we are able to provide diverse learners with relatable learning contexts and meaningful activities.

We ensure minorities see themselves in the industry and learning environment. Our courses have a diverse cast of characters from varied cultures, countries, and families. At Code Avengers, we support and encourage people from all walks of life to get excited about being creators and innovators in the digital space. The majority of our content is available in English and Spanish, with a growing library of content in German, French, Japanese, Maori, Dutch, Portuguese, Russian, Chinese/Mandarin, and other languages planned as well. Code Avengers has been used by over 2 million students worldwide.

Promoting Learning

At its core, CS is about solving problems. Digital outcomes, products, and services are developed in order to find innovative solutions to problems and opportunities we face. That's why at Code Avengers, we integrate problem solving into every learning resource.

Divergent thinking is supported in courses that require students to generate original ideas and examine a variety of viable solutions. Students can investigate tasks in a 'non-linear' manner using logic puzzles that explicitly teach decision-making and problem-solving processes on our platform. We aim to create an environment in which students can gain firsthand experience with the concepts of optimization, redundancy, and reliability, as well as their implications for technological system design, development, and maintenance.

Individualized Learning

Teachers love the self-paced Code Avengers curriculum as it encourages and enables students to be confident, independent learners. Our courses are perfect for students of any experience level and have many tools available for students with diverse learning needs. When designing courses, we use a variety of strategies to support students and ensure learning is accessible. All tasks work similarly and have a predictable layout, and audio is available for most courses to help reduce literacy barriers.

Students enjoy the Code Avengers platform because of all the features it offers including immediate feedback, helpful videos, pop-up vocabulary, hints, a built-in integrated development environment (IDE), achievement tracking, and other resources. All tasks are self-graded with real-time progress tracking provided on the teacher dashboard and class data can be exported to a spreadsheet at any time. Teachers can maintain control of class direction by assigning specific modules to class groups or individual students and simultaneously manage class differentiation needs.

Not only can our content be tailored to our students, but our teachers also benefit from the platform's adaptability. Teachers can learn, explore, and upskill alongside their students without having to be experts. Code Avengers makes it easy for teachers with any amount of experience to facilitate effective CS learning in their classrooms. It is a high priority of ours to provide teachers with a suite of resources to support the delivery and monitoring of student learning, including live student tracking, analytics, lesson plans, quizzes, offline activities, and more. We save teachers time while providing clear scaffolding, scope, and sequence. This is truly what separates us from the rest; Code Avengers supports TEACHERS.

Collaboration

Unplugged activities aimed at helping students think critically and strategically are available to support many of the courses. Students take on specific roles typical in a development cycle to contribute to a project as a team and are taught to manage their time and resources to produce digital artifacts.

We also provide support and guidance for teachers that includes effective pedagogy and teaching strategies to promote collaboration in Computer Science learning.

Collaboration is fundamental to our internal ethos. We encourage learners and teachers to provide feedback and make suggestions for how to improve our platform, and we actively seek both perspectives when identifying new resources for development.

Real-World Applications

Our courses offer authentic learning opportunities that propose real-world challenges. This allows students to apply their understanding of CS concepts through memorable and compelling applications. Students are expected to use higher-order cognitive processes to analyze problems using a range of skills, such as detail-oriented coding, teamwork, and big-picture, long-term thinking.

Our Junior courses are story-based, and the learning happens within relevant real-world scenarios. At the Pro (senior) level, students write code to build real-world programs and apps that have actual everyday applications.

The impacts of computing are an important topic addressed in many of our courses. Students design and iteratively develop computational artifacts for practical purposes, personal expression, and to address societal issues. They are required to critically evaluate computational artifacts and consider both their beneficial and harmful effects on society. At higher levels, they theorize about the future of the computing innovations that have altered our culture and discuss the legal and ethical implications of these innovations.

Career & Technical Education

Students transition to learning industry-standard programming languages, such as JavaScript and Python, and web languages such as HTML5 and CSS3, or build their theoretical knowledge with our Computer Science and Design courses. These are some of the most in-demand languages in the tech sector, whilst also being very beginner-friendly for both students and teachers. This means learners are given an excellent foundation to move into entry-level/internship-type roles, or on to further study.

Cross-Curricular Connections

Cross-curricular links help learners to make connections with a range of other domains including further STEM topics, social sciences, and the arts. As a company that is already well established in the field of EdTech, Code Avengers is especially well placed to continue to adapt our product to enhance cross-curricular learning. This positions us well to support schools and teachers with the introduction of CS learning as curricula begin to roll out across various states of the USA. We understand that teachers sometimes need to work CS into their classrooms in intelligent ways, so our courses are built to integrate into mainstream learning and connect concepts across the curriculum. When developing new content, cross-curricular connections are a fundamental part of the planning process. Some examples of cross-curricular links found in our existing content are:

- Design theory e.g. color theory.
- History e.g. the history of numbering systems, the evolution of agriculture with AI and agritech.
- Social sciences e.g. the ethics and social impacts of current and future technology.
- Language & literacy e.g. writing and literature-themed activities and writing prompts, procedural writing, constructing texts for specific audiences.
- Science e.g. the physics of light and sound.
- Math e.g. geometry, coordinates, statistics, and more.

Research & Case Studies

Queensland Pilot

In late 2019, a trial was conducted with Trinity Bay State High School in Manunda, Queensland Australia. It has a demographic of approximately 35% indigenous learners. Having previously had difficulty engaging their learners with digital technologies, the school piloted the Code Avengers platform to explore whether it would improve student engagement and learning outcomes.

The trial was a resounding success, with student engagement significantly improved, particularly for indigenous girls. The pilot report can be read here, summarized in the quote below with key findings and benefits of using Code Avengers:

> “Differentiation of Digital Technologies course content is now truly possible with the Code Avengers platform. We have a very large Special Needs Program at the school with over 150 students identified with special needs/cognitive disorders/physical issues. Many of these students have worked extremely effectively on the Code Avengers platform. This is an area where their various disabilities are of little consequence. Many of the specialist staff that support them have been extremely surprised at the results that the students have achieved using Code Avengers.”

Chile Pilot

In 2020-2021, a tertiary education provider in Chile ran a pilot program using the Code Avengers platform for their learners. The institution had previously struggled with a failure rate of over 30% for their Foundations of Computer Science course, which covered topics such as the basics of computer programming and web development.

A number of campuses were given access to the platform as well as guidance on which courses learners should complete to support their coursework. A threshold of 2000 points was established, representing the Code Avengers courses best aligned with a particular segment of their coursework. Of the students that met or exceeded the 2000-point threshold, 86% passed the course. This meant a 14% failure rate–less than half of the previous 30+% failure rate. There was a strong correlation, therefore, between learner engagement with the Code Avengers platform, and pass rates.

Chicago Case Study

In 2012, the Chicago Public Schools recognized a discrepancy: the fastest-growing job sector, Information Technology, was predominantly white and overwhelmingly male. They were determined to shift that balance.

When the Chicago Public Schools began to pilot an "Exploring Computer Science" course several years ago, the staff at Whitney Young High School jumped on board. Their primary goal was to increase diversity in the computer science field - more women, more people of color. The innovation took hold and before long, the staff at Whitney Young determined that they needed more - more content, more lesson plans, and more assessments. They wanted to take computer science instruction to the next level.

One of the major reasons why CS gained so much traction at Whitney Young was due to them adopting Code Avengers as a tool. Code Avengers helps break down the stereotypes around Computer Science. Students at Whitney Young saw that it was not just the stereotypical math-and-science-minded individuals who can succeed at Computer Science. Many of the students who took the introductory Computer Science course gained confidence and continued on to explore advanced coursework including AP level Computer Science courses. This opened up more opportunities for students to challenge themselves with college-level Computer Science instruction.

Testimonials

The University of Auckland:

> "Our course is a core component of the Bachelor's Degree of Commerce, with well over 2,000 students enrolled each year. Providing personal training in coding, along with helpful evaluation and feedback for each student in such a course is extremely difficult, if not impossible. Since 2014, codeavengers.com has been supporting our course with its outstanding products, including suitable coding lessons for each individual student and fantastic technical support. The teaching team has been receiving positive feedback from the class about the learning environment on the website, and how it encourages students to further explore different aspects of programming. On the other hand, the student evaluation functionalities on codeavengers.com made it so much easier to mark and monitor students' progress. Thanks to the team at codeavengers.com, our faculty now has a course that can effectively and efficiently introduce large groups of students to the amazing world of programming."

David (Head of Department, Tawa College, NZ):

> "**What students like:** We have received positive feedback about the online courses. Students can learn in their own time at their own pace. Nice simple interface, not too much text for instructions with help functions. Also, the level of learning aligns really well with what is expected at years 11, 12, and 13 from NZQA [New Zealand Qualifications Authority].
>
> **What teachers like:** We have appreciated the fact that the courses are at a good level and align with the NZQA Assessments that we set for our students. Examples; Python 1, 2, and 3 all scaffold nicely with students completing the online course for homework during the Term, then as teachers we recap the learning on CA in class so students are ready for their final NZQA / NCEA [National Certificate in Educational Achievement] Assessment."

Tatiana (Independent Learner):

> "It is my first time using code avengers. It is a really well-structured page and I enjoy it. I have never seen a code-teaching website that is this good. The rest that I have seen are mediocre at best. Although Khan Academy had some interesting stuff. But I 100% prefer code avengers and I will continue to learn using it."

Aimee (Teacher, Sage School, MA):

> "Code Avengers is the perfect platform for kids to gain fluency in programming languages. We serve gifted students so the option for kids to independently advance ahead in lessons is crucial."

Whitney Young High School (Chicago, IL)

> "Student work is self-paced, monitored easily through the progress monitoring dashboard, and teachers can use that data to support student learning with one-on-one or small group instruction, pairing students for peer coaching, or offering additional challenges for those who master content quickly.
>
> Code Avengers delivers everything necessary for a school to equip students with the digital technology skills that will start them on the path to be primed to apply for (and land) the computer-science-based jobs of today and tomorrow: a wide range of Computer Science content, pedagogically sound ready-to-use lesson plans, solid teacher support, progress monitoring, engaging and scaffolded lesson plans, and on-target assessments."

Andrew Mauer-Oats (Whitney Young High School):

> "Code Avengers gives my advanced students the chance to run on ahead; nothing holds them back from learning all that they wish to learn and the platform is loaded with tons of content giving my students the chance to learn HTML, CSS, or Python."

Indicative Course Map

Fundamentals of Computer Science

Introduction to Python PYTHON 0 PRO	Robot Restaurant Designer COMPUTATIONAL THINKING 300 JR
Larsson Castle Mystery PROGRAMMING 3 JR	Saving Food Avengers COMPUTATIONAL THINKING 3 PRO
Shield Design Showdown PROGRAMMING 300 JR	Security Siege IMPACTS OF COMPUTING 300 JR
Blockly Flags JAVASCRIPT 123 PRO	Jumping Jam DIGITAL MEDIA 300 JR
AVATAR: Big Data & Digital Footprints IMPACTS OF COMPUTING 600 JR	Sequence Grid Challenge JAVASCRIPT 101 PRO
Operation Cloud NETWORKS & SECURITY 600 JR	Drawing Flags JAVASCRIPT 103 PRO
The Information Transformation DATA & ANALYSIS 60 JR	Game Dev: Food Frenzy JAVASCRIPT 105 PRO

Computer Science I

Digital Postcard HTML/CSS 101 PRO	Wires, Waves, & Wifi DIGITAL INFRASTRUCTURE 4 JR
Robot Restaurant Puzzler ALGORITHMS 0 PRO	Mind vs. Machine PROGRAMMING 4 JR
Intro to Graphics (Tangrams) JAVASCRIPT 102 PRO	AI, AI, Drones! IMPACTS OF COMPUTING 400 JR
Introduction to JavaScript JAVASCRIPT 0 PRO	Game Dev: Sneaky Ninja JAVASCRIPT 110 PRO
Color Conundrum DATA REPRESENTATION 400 JR	Photo Booth WEB DEVELOPMENT 100 PRO
Game Dev: Pixel Robbers JAVASCRIPT 100 PRO	Introduction to Web Development WEB DEVELOPMENT PRO
Intro to Python Turtle Graphics PYTHON 100 PRO	Introduction to jQuery JQUERY 0 PRO

Computer Science II

Algorithms ALGORITHMS 1 PRO	Planning for Programming SOFTWARE ENGINEERING 1 PRO
How the Web Works DIGITAL SYSTEMS 1 PRO	Variables, Else/If & Loops PYTHON 1 PRO
Introduction to HTML/CSS HTML/CSS 0 PRO	Variables, If Statements, & Loops JAVASCRIPT 1 PRO
Typography DESIGN 1 PRO	Build a Website HTML/CSS 1 PRO
Color Theory DESIGN 2 PRO	JavaScript Interactivity WEB DEVELOPMENT 1 PRO
Numbers Through Time DATA REPRESENTATION 3 JR	ReconFigure of Speech PROGRAMMING LANGUAGES 100 PRO
Human-Computer Interaction HCI 1 PRO	Reality Bytes DATA REPRESENTATION 4 JR

Computer Science III

High- & Low-Level Languages PROGRAMMING LANGUAGES 1 PRO	**For Python, JavaScript, Web Development and HTML-CSS complete Level 1, then 2, before doing level 3.*
Lists and Functions PYTHON 2 PRO	GUI & Object-Oriented Python PYTHON 3 PRO
Strings, Functions, Arrays, & Math JAVASCRIPT 2 PRO	Events, Animation, GUIs, & Classes JAVASCRIPT 3 PRO
Build an Interactive Guide HTML/CSS 2 PRO	Build a Responsive Blog Site HTML/CSS 3 PRO
Loops, Timers, and Dynamic Firebase Apps WEB DEVELOPMENT 2 PRO	Build a Real-Time Chat App with Firebase WEB DEVELOPMENT 3 PRO
jQuery Basics JQUERY 1 PRO	

Background & Detail

Each of the 10 subtopics contains a number of performance indicators. In total there are 190 performance indicators across four levels. We closely align to 152 indicators, partly align to 23, and do not align to 15 topics.

In its current state, our content aligns particularly well with Fundamentals of CS and Computer Science I, and aligns well with Computer Science II and Computer Science III in Python and JavaScript.

Some of the performance indicators are written in a way that is specific to the Java programming language, which we don't cover. For this reason, we've provided the key below, for the following full lists of performance indicators, showing where some indicators are not relevant, or we cover the equivalent in Python/JavaScript.

Key

Color	Explanation
🟩	Code Avengers content covers this standard
**	Code Avengers content may cover part of this standard, but it is knowledge or skills that would be covered organically in class or school as a result of participating in Computer Science learning e.g. careers advice, university programs, file management on devices, team work, leadership skills, and so on.
*	The standards are written for the Java language. Code Avengers doesn't offer Java at present, however it does offer Python and JavaScript-both of which are more commonly used as beginner languages these days due to them being simpler to learn and having more straightforward syntax. These standards are either not relevant in Python or JavaScript, or the equivalent Python or JavaScript concept is covered.
🟨	Code Avengers content partially covers this standard.
🟥	Code Avengers content does not cover any of this standard at present.

Fundamentals of Computer Science

Students shall be awarded one-half to one credit for successful completion of this course. The prerequisite for this course is proficiency in the knowledge and skills relating to Technology Applications, Grades 6-8. This course is recommended for students in Grades 9-12.

- Fundamentals of Computer Science is intended as a first course for those students just beginning the study of computer science.
- Students will develop creativity and innovation by solving real-world problems.
- Students will collaborate to access, analyze, and evaluate problem-solving information using computer science.
- Students will learn the problem-solving and reasoning skills that are the foundation of computer science.
- Students will select the appropriate technology, synthesize knowledge, create solutions, and evaluate results using computer science knowledge and skills.
- Students will learn digital citizenship by researching and practicing current laws and regulations.

Fundamentals of Computer Science

1. Employability

1.1.	The student identifies various employment opportunities in the computer science field. The student is expected to:	
1.1.1.	Identify and demonstrate employable work behaviors such as regular attendance, punctuality, maintenance of a professional work environment, and effective written and verbal communication	**
1.1.2.	Understand the denary, binary and hexadecimal number systems identify and demonstrate positive personal qualities such as authenticity, resilience, initiative, and a willingness to learn new knowledge and skills	🟩
1.1.3.	Solve problems and think critically	🟩
1.1.4.	Demonstrate leadership skills and function effectively as a team member	**
1.1.5.	Demonstrate an understanding of ethical and legal responsibilities and ramifications in relation to the field of cybersecurity.	🟨

2. Creativity and innovation

2.1.	The student develops products and generates new knowledge, understanding, and skills. The student is expected to:	
2.1.1.	Investigate and explore various career opportunities within the computer science field and report findings through various media	**
2.1.2.	Create algorithms for the solution of various problems	🟩
2.1.3.	Discuss methods and create and publish web pages using a web-based language such as HTML, Java Script, or XML	🟩
2.1.4.	Use generally accepted design standards for spacing, fonts, and color schemes to create functional user interfaces, including static and interactive screens.	🟩

3. Communication and collaboration

3.1.	The student communicates and collaborates with peers to contribute to his or her own learning and the learning of others. The student is expected to:	
3.1.1.	Seek and respond to advice or feedback from peers, educators, or professionals when evaluating problem solutions;	🟩
3.1.2.	Debug and solve problems using reference materials and effective strategies	🟩

3.1.3.	Publish information in a variety of ways such as print, monitor display, web pages, or video.	🟩
4.	**Critical thinking, problem solving, and decision making.**	
4.1.	The student uses appropriate strategies to analyze problems and design algorithms. The student is expected to:	
4.1.1.	Demonstrate the ability to insert external standalone objects such as scripts or widgets into web pages;	🟩
4.1.2.	Communicate an understanding of binary representation of data in computer systems, perform conversions between decimal and binary number systems, and count in binary number systems	🟩
4.1.3.	Identify a problem's description, purpose, and goals	🟩
4.1.4.	Demonstrate coding proficiency in a programming language by developing solutions that create stories, games, and animations	🟩
4.1.5.	Identify and use the appropriate data type to properly represent the data in a program problem solution	🟩
4.1.6.	Demonstrate coding proficiency in a programming language by developing solutions that create stories, games, and animations	🟩
4.1.7.	Communicate an understanding of and use variables within a programmed story, game, or animation	🟩
4.1.8.	Uuse arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division;	🟩
4.1.9.	Communicate an understanding of and use sequence within a programmed story, game, or animation	🟩
4.1.10.	Communicate an understanding of and use conditional statements within a programmed story, game, or animation	🟩
4.1.11.	Communicate an understanding of and use iteration within a programmed story, game, or animation	🟩
4.1.12.	Use random numbers within a programmed story, game, or animation	🟩
4.1.13.	Test program solutions by investigating intended outcomes	🟩
5.	**Digital citizenship**	
5.1.	The student understands and demonstrates the social responsibility of end users regarding significant issues related to digital technology, digital hygiene, and cyberbullying. The student is expected to:	

5.1.1.	Identify and understand the nature and value of privacy	🟩
5.1.2.	Analyze the positive and negative implications of a digital footprint and the maintenance and monitoring of an online presence	🟩
5.1.3.	Discuss the role and impact of technology on privacy	🟩
5.1.4.	Identify the signs, emotional effects, and legal consequences of cyberbullying and cyberstalking	🟨
5.1.5.	Identify and discuss effective ways to deter and report cyberbullying	🟨
5.1.6.	Analyze how electronic media can affect reliability of information	🟩
6.	**Technology operations and concepts**	
6.1.	The student understands technology concepts, systems, and operations as they apply to computer science. The student is expected to:	
6.1.1.	Identify and explain the function of basic computer components, including a central processing unit (CPU), storage, and peripheral devices	🟩
6.1.2.	Use system tools, including appropriate file management	**
6.1.3.	Compare different operating systems	🟩
6.1.4.	Describe the differences between an application and an operating system	🟩
6.1.5.	Use various input, processing, output, and primary/secondary storage devices	**

Computer Science 1

Students shall be awarded one-half to one credit for successful completion of this course. The required prerequisite for this course is Algebra I. This course is recommended for students in Grades 9-12.

- Computer Science I is designed to foster students' creativity and innovation by presenting opportunities to design, implement and present meaningful programs through a variety of media.
- Students will collaborate with others, the instructor, and online communities to solve course problems.
- Data analysis includes identifying task requirements, planning search strategies, and using computer science concepts to access, analyze, and evaluate problem-solving information.
- Students will select the appropriate technology, synthesize knowledge, create a solution, and evaluate the results using computer science knowledge and skills.
- Students will practice integrity and respect throughout Computer Science I.
- Technology operations, systems, and concepts help students understand computer science principles.

Computer Science I

1.	**Employability**	
1.1.	The student identifies various employment opportunities in the computer science field. The student is expected to:	
1.1.1.	Identify job and internship opportunities and accompanying job duties and tasks and contact one or more companies or organizations to explore career opportunities	**
1.1.2.	Examine the role of certifications, resumes, and portfolios in the computer science profession	
1.1.3.	Employ effective technical reading and writing skills	
1.1.4.	Employ effective verbal and non-verbal communication skills	
1.1.5.	Solve problems and think critically	
1.1.6.	Demonstrate leadership skills and function effectively as a team member	**
1.1.7.	Communicate an understanding of legal and ethical responsibilities in relation to the field of computer science	
1.1.8.	Demonstrate planning and time-management skills	
1.1.9.	Compare university computer science programs	**
2.	**Communication and collaboration**	
2.1.	The student communicates and collaborates with peers to contribute to his or her own learning and the learning of others. The student is expected to:	
2.1.1.	Participate in learning communities as a learner, initiator, contributor, and teacher/mentor	
2.1.2.	Seek and respond to advice from peers, educators, or professionals when evaluating quality and accuracy of the student's product	
3.	**Programming style and presentation**	
3.1.	The student utilizes proper programming style and develops appropriate visual presentation of data, input, and output. The student is expected to:	
3.1.1.	Create and properly label and display output	

3.1.2.	Create interactive input interfaces, with relevant user prompts, to acquire data from a user such as console displays or Graphical User Interfaces (GUIS)	🟩
3.1.3.	Write programs with proper programming style to enhance the readability and functionality of a code by using descriptive identifiers, internal comments, white space, spacing, indentation, and a standardized program style	🟩
3.1.4.	Format data displays using standard formatting styles; and (E) display simple vector graphics using lines, circles, and rectangles	🟩
4.	**Critical thinking, problem-solving, and decision-making**	
4.1.	The student uses appropriate strategies to analyze problems and design algorithms. The student is expected to:	
4.1.1.	Use program design problem-solving strategies such as flowchart or pseudocode to create program solutions	🟩
4.1.2.	Create a high-level program plan using a visual tool such as a flowchart or graphic organizer	🟩
4.1.3.	Identify the tasks and subtasks needed to solve a problem	🟩
4.1.4.	Identify the data types and objects needed to solve a problem	🟩
4.1.5.	Identify reusable components from existing code	🟩
4.1.6.	Design a solution to a problem	🟨
4.1.7.	Code a solution from a program design	🟩
4.1.8.	Identify error types, including syntax, lexical, run time, and logic	🟩
4.1.9.	Test program solutions with valid and invalid test data and analyze resulting behavior	🟩
4.1.10.	Debug and solve problems using error messages, reference materials, language documentation, and effective strategies	🟩
4.1.11.	Create and implement common algorithms such as finding greatest common divisor, finding the biggest number out of three, finding primes, making change, and finding the average	🟩
4.1.12.	Create program solutions that address basic error handling such as preventing division by zero and type mismatch	🟩
4.1.13.	Select the most appropriate construct for a defined problem	🟨
4.1.14.	Create program solutions by using the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division	🟩

4.1.15.	Create program solutions to problems using available mathematics library functions or operators, including absolute value, round, power, square, and square root	🟩
4.1.16.	Develop program solutions that use assignment	🟩
4.1.17.	Develop sequential algorithms to solve non-branching and non-iterative problems	🟩
4.1.18.	Develop algorithms to decision-making problems using branching control statements	🟩
4.1.19.	Develop iterative algorithms and code programs to solve practical problems	🟩
4.1.20.	Demonstrate the appropriate use of the relational operators	🟩
4.1.21.	Demonstrate the appropriate use of the logical operators	🟩
4.1.22.	Generate and use random numbers	🟩
5.	**Digital citizenship**	
5.1.	The student explores and understands safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:	
5.1.1.	Discuss and explain intellectual property, privacy, sharing of information, copyright laws, and software licensing agreements	🟨
5.1.2.	Practice ethical acquisition and use of digital information	**
5.1.3.	Demonstrate proper digital etiquette, responsible use of software, and knowledge of acceptable use policies	**
5.1.4.	Investigate privacy and security measures, including strong passwords, pass phrases, and other methods of authentication and virus detection and prevention	🟩
5.1.5.	Investigate computing and computing-related advancements and the social and ethical ramifications of computer usage	🟨
6.	**Technology operations, systems, and concepts**	
6.1.	The student understands technology concepts, systems, and operations as they apply to computer science. The student is expected to:	
6.1.1.	Identify and describe the function of major hardware components, including primary and secondary memory, a central processing unit (CPU), and peripherals	**
6.1.2.	Differentiate between current programming languages, discuss the general purpose for each language, and demonstrate knowledge of	🟩

	specific programming terminology and concepts and types of software development applications	🟩
6.1.3.	Differentiate between a high-level compiled language and an interpreted language	🟩
6.1.4.	Identify and use concepts of object-oriented design	🟩
6.1.5.	Differentiate between local and global scope access variable declarations	🟩
6.1.6.	Encapsulate data and associated subroutines into an abstract data type	🟥
6.1.7.	Create subroutines that do not return values with and without the use of arguments and parameters	🟩
6.1.8.	Create subroutines that return typed values with and without the use of arguments and parameters	🟩
6.1.9.	Create calls to processes passing arguments that match parameters by number, type, and position	🟩
6.1.10.	Compare data elements using logical and relational operators	🟩
6.1.11.	Identify and convert binary representation of numeric and nonnumeric data in computer systems using American Standard Code for Information Interchange (ASCII) or Unicode	🟩
6.1.12.	Identify finite limits of numeric data such as integer wrap around and floating point precision	🟨
6.1.13.	Perform numerical conversions between the decimal and binary number systems and count in the binary number system	🟩
6.1.14.	Choose, identify, and use the appropriate data types for integer, real, and Boolean data when writing program solutions	🟩
6.1.15.	Analyze the concept of a variable, including primitives and objects	🟩
6.1.16.	Represent and manipulate text data, including concatenation and other string functions	🟩
6.1.17.	Identify and use the structured data type of one-dimensional arrays to traverse, search, and modify data	🟩
6.1.18.	Choose, identify, and use the appropriate data type or structure to properly represent the data in a program problem solution	🟩
6.1.19.	Compare strongly typed and untyped programming languages	🟥

Computer Science II

Students shall be awarded one credit for successful completion of this course. The required prerequisites for this course are Algebra I and either Computer Science I or Fundamentals of Computer Science. This course is recommended for students in Grades 11 and 12.

- Computer Science II will foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media.
- Students will work with their instructor, classmates, and online communities to solve course problems.
- Students will use data analysis to identify task requirements, plan search strategies, and access, analyze, and evaluate problem-solving information.
- Students will select the appropriate technology, synthesize knowledge, create solutions, and evaluate results using computer science knowledge and skills.
- Students will learn digital citizenship by researching and practicing current laws and regulations.
- Technology operations, systems, and concepts help students understand computer science.

Computer Science II		
1.	**Employability**	
1.1.	The student understands technology concepts, systems, and operations as they apply to computer science. The student is expected to:	
1.1.1.	Identify job and internship opportunities and accompanying job duties and tasks and contact one or more companies or organizations to explore career opportunities	**
1.1.2.	Examine the role of certifications, resumes, and portfolios in the computer science profession	🟩
1.1.3.	Employ effective technical reading and writing skills	🟩
1.1.4.	Employ effective verbal and non-verbal communication skills	🟩
1.1.5.	Solve problems and think critically	🟩
1.1.6.	Demonstrate leadership skills and function effectively as a team member	**
1.1.7.	Identify legal and ethical responsibilities in relation to the field of computer science	🟩
1.1.8.	Demonstrate planning and time-management skills	🟩
1.1.9.	Compare university computer science programs	**

2.	**Creativity and innovation**	
2.1.	The student develops products and generates new understandings by extending existing knowledge. The student is expected to:	
2.1.1.	Use program design problem-solving strategies to create program solutions	🟩
2.1.2.	Read, analyze, and modify programs and their accompanying documentation such as an application programming interface (API), internal code comments, external documentation, or readme files	🟨
2.1.3.	Follow a systematic problem-solving process that identifies the purpose and goals, the data types and objects needed, and the subtasks to be performed	🟩
2.1.4.	Compare design methodologies and implementation techniques such as top-down, bottom-up, and black box	🟥
2.1.5.	Trace a program, including inheritance and black box programming	🟥
2.1.6.	Choose, identify, and use the appropriate abstract data type, advanced data structure, and supporting algorithms to properly represent the data in a program problem solution	🟩
2.1.7.	Use object-oriented programming development methodology, including data abstraction, encapsulation with information hiding, inheritance, and procedural abstraction in program development.	🟩
3.	**Communication and collaboration**	
3.1.	The student communicates and collaborates with peers to contribute to his or her own learning and the learning of others. The student is expected to:	
3.1.1.	Use the principles of software development to work in software design teams	**
3.1.2.	Break a problem statement into specific solution requirements	🟩
3.1.3.	Create a program development plan	🟩
3.1.4.	Code part of a solution from a program development plan while a partner codes the remaining part	🟨
3.1.5.	Collaborate with a team to test a solution, including boundary and standard cases	**
3.1.6.	Develop presentations to report the solution findings	**
4.	**Data literacy and management**	
4.1.	The student locates, analyzes, processes, and organizes data. The student is expected to:	

4.1.1.	Use programming file structure and file access for required resources	🟩
4.1.2.	Acquire and process information from text files, including files of known and unknown sizes	🟩
4.1.3.	Manipulate data using string processing	🟩
4.1.4.	Manipulate data values by casting between data types	🟩
4.1.5.	Use the structured data type of one-dimensional arrays to traverse, search, modify, insert, and delete data	🟩
4.1.6.	Identify and use the structured data type of two-dimensional arrays to traverse, search, modify, insert, and delete data	🟩
4.1.7.	Identify and use a list object data structure to traverse, search, insert, and delete data	🟩
4.1.8.	Differentiate between categories of programming languages, including machine, assembly, high-level compiled, high-level interpreted, and scripted	🟩
5.	**Critical thinking, problem solving, and decision making**	
5.1.	The student explores and understands safety, legal, cultural, and societal issues relating to the use of technology and information. The student is expected to:	
5.1.1.	Develop sequential algorithms using branching control statements, including nested structures, to create solutions to decision-making problems	🟩
5.1.2.	Develop choice algorithms using selection control statements based on ordinal values	🟩
5.1.3.	Demonstrate the appropriate use of short-circuit evaluation in certain situations	🟥
5.1.4.	Use Boolean algebra, including De Morgan's Law, to evaluate and simplify logical expressions	🟥
5.1.5.	Develop iterative algorithms using nested loops;	🟩
5.1.6.	Identify, trace, and appropriately use recursion in programming solutions, including algebraic computations	🟨
5.1.7.	Trace, construct, evaluate, and compare search algorithms, including linear searching and binary searching	🟥
5.1.8.	Identify, describe, trace, evaluate, and compare standard sorting algorithms, including selection sort, bubble sort, insertion sort, and merge sort	🟥

5.1.9.	Measure time and space efficiency of various sorting algorithms, including analyzing algorithms using "big-O" notation for best, average, and worst-case data patterns	🟥
5.1.10.	Develop algorithms to solve various problems such as factoring, summing a series, finding the roots of a quadratic equation, and generating Fibonacci numbers	🟨
5.1.11.	Test program solutions by investigating boundary conditions; testing classes, methods, and libraries in isolation; and performing stepwise refinement	🟩
5.1.12.	Identify and debug compile, syntax, runtime, and logic errors	🟩
5.1.13.	Compare efficiency of search and sort algorithms by using informal runtime comparisons, exact calculation of statement execution counts, and theoretical efficiency values using "big-O" notation, including worst-case, best-case, and average-case time/space analysis	🟥
5.1.14.	Count, convert, and perform mathematical operations in the decimal, binary, octal, and hexadecimal number systems	🟩
5.1.15.	Identify maximum integer boundary, minimum integer boundary, imprecision of real number representations, and round-off errors	🟨
5.1.16.	Create program solutions to problems using a mathematics library	🟨
5.1.17.	Use random number generator algorithms to create simulations	🟩
5.1.18.	Use composition and inheritance relationships to identify and create class definitions and relationships	🟨
5.1.19.	Explain and use object relationships between defined classes, abstract classes, and interfaces	🟨
5.1.20.	Create object-oriented class definitions and declarations using variables, constants, methods, parameters, and interface implementations	🟩
5.1.21.	Create adaptive behaviors using polymorphism	🟥
5.1.22.	Use reference variables for object and string data types	🟨

Computer Science III

Students shall be awarded one credit for successful completion of this course. The required prerequisites for this course are Algebra I and either Computer Science I or Fundamentals of Computer Science. This course is recommended for students in Grades 11 and 12.

- Computer Science II will foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media.

- Students will work with their instructor, classmates, and online communities to solve course problems.
- Students will use data analysis to identify task requirements, plan search strategies, and access, analyze, and evaluate problem-solving information.
- Students will select the appropriate technology, synthesize knowledge, create solutions, and evaluate results using computer science knowledge and skills.
- Students will learn digital citizenship by researching and practicing current laws and regulations.
- Students will gain an understanding of computer science through the study of technology operations, systems, and concepts

Computer Science III		
1.	**Employability**	
1.1.	The student identifies various employment opportunities in the computer science field. The student is expected to:	
1.1.1.	Identify job and internship opportunities and accompanying job duties and tasks and contact one or more companies or organizations to explore career opportunities	**
1.1.2.	Examine the role of certifications, resumes, and portfolios in the computer science profession	🟩
1.1.3.	Employ effective technical reading and writing skills	🟩
1.1.4.	Employ effective verbal and non-verbal communication skills	🟩
1.1.5.	Solve problems and think critically	🟩
1.1.6.	Demonstrate leadership skills and function effectively as a team member	**
1.1.7.	Demonstrate an understanding of legal and ethical responsibilities in relation to the field of computer science	🟩
1.1.8.	Demonstrate planning and time-management skills	🟩
1.1.9.	Compare university computer science programs	**
2.	**Creativity and innovation**	
2.1.	The student develops products and generates new understandings by extending existing knowledge. The student is expected to:	

2.1.1.	Apply object-oriented programming, including data abstraction, encapsulation, inheritance, and polymorphism, to manage the complexity of a project	🟩
2.1.2.	Design and implement a class hierarchy	🟩
2.1.3.	Read and write class specifications using visual organizers, including Unified Modeling Language	🟨
2.1.4.	Identify, describe, evaluate, compare, and implement standard sorting algorithms that perform sorting operations on data structures, including quick sort and heap sort	🟥
2.1.5.	Identify and use the appropriate abstract data type, advanced data structure, and supporting algorithms to properly represent the data in a program problem solution	🟩
3.	**Communication and collaboration**	
3.1.	The student communicates and collaborates with peers to contribute to his or her own learning and the learning of others. The student is expected to:	
3.1.1.	Use networked tools for file management and collaboration	**
3.1.2.	Work in software design teams.	🟨
4.	**Data literacy and management**	
4.1.	The student locates, analyzes, processes, and organizes data. The student is expected to:	
4.1.1.	Identify and use two-dimensional ragged arrays to traverse, search, modify, insert, and delete data	*
4.1.2.	Describe and demonstrate proper linked list management, including maintaining the head and safe addition and deletion of linked objects	*
4.1.3.	Create or trace program solutions using a linked-list data structure, including unordered single, ordered single, double, and circular linked	*
4.1.4.	Describe composite data structures, including a linked list of linked lists	*
4.1.5.	Create or trace program solutions using stacks, queues, trees, heaps, priority queues, graph theory, and enumerated data types	*
4.1.6.	Create or trace program solutions using sets, including hash and tree-based data structures	🟩
4.1.7.	Create or trace program solutions using map style data structures	🟩
4.1.8.	Write and modify text file data	🟩
5.	**Critical thinking, problem solving, and decision making**	

5.1.	The student uses appropriate strategies to analyze problems and design algorithms. The student is expected to:	
5.1.1.	Evaluate expressions using bitwise operators	🟥
5.1.2.	Evaluate expressions using the ternary operator	🟩
5.1.3.	Identify, trace, and appropriately use recursion in programming solutions, including processing binary trees	🟥
5.1.4.	Create or trace program solutions using hashing	🟥
5.1.5.	Explore common algorithms such as matrix addition and multiplication, fractals, Towers of Hanoi, and magic square	🟨
5.1.6.	Create program solutions that exhibit robust behavior by recognizing and avoiding runtime errors and handling anticipated errors	🟩
6.	**Testing and documentation**	
6.1.	The student demonstrates appropriate documentation and testing practices. The student is expected to:	
6.1.1.	Use appropriate formatting and write documentation to support code maintenance, including pre- and post-condition statements	🟩
6.1.2.	Write program assumptions in the form of assertions	*
6.1.3.	Write a Boolean expression to test a program assertion	*
6.1.4.	Construct assertions to make explicit program invariants	*
7.	**Practical application of technology**	
7.1.	The student utilizes technology concepts, systems, and operations as they apply to computer science. The student is expected to:	
7.1.1.	Analyze and create computer program workflow charts and basic system diagrams, documenting system functions, features, and operations	🟨
7.1.2.	Gather requirements, design, and implement a process by which programs can interact with each other such as using interfaces	🟨
7.1.3.	Create simple programs using a low-level language such as assembly	🟨
7.1.4.	Create discovery programs in a high-level language	🟩
7.1.5.	Create scripts for an operating system;	🟥
7.1.6.	Explore industry best practices for secure programming; and	🟨
7.1.7.	Explore emerging industry or technology trends	🟩