GRADE 2 UNIT PLAN

CODE AVENGERS

Algorithms & Programming | **Fishful Thinking** | 10 hrs

Unit Description

In this unit, students will engage in a series of hands-on activities and collaborative projects to explore coding and creativity using Blockly and Scratch. They will start by familiarizing themselves with the Blockly visual programming language, using drag-and-drop blocks to create code, and later experimenting with different blocks in a freeplay scene. This unit will use the **PR1: Stormy Day** Code Avengers course.

Learning Outcomes

Target ages: US/CA Grade 2 | **AU** Year 2 | **NZ/UK** Year 3.

- Program an animated text (story, song, or rap) using blockly.
- Describe a program's sequence of events and predict expected outcomes.
- Analyze stories and retell events in the correct order.
- Manipulate numbers to solve word problems.
- Create a program for an animated text (story, song, or rap) using Blockly.

Curriculum Links

US:

- **1A-AP-08:** Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.
- **1A-AP-10:** Develop programs with sequences and simple loops, to express ideas or address a problem.
- **1A-AP-12:** Develop plans that describe a program's sequence of events, goals, and expected outcomes.
- **1A-AP-13:** Give attribution when using the ideas and creations of others while developing programs.
- **1A-AP-14:** Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.
- **1A-AP-15:** Using correct terminology, describe steps taken and choices made during the iterative process of program development.

UK:

- **KS1 CO1:** Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions.

AU:

- **AC9TDI2P02:** Follow and describe algorithms involving a sequence of steps, branching (decisions) and iteration (repetition).
- **AC9TDI2P04:** Use the basic features of common digital tools to create, locate and communicate content.
- **AC9TDI2P05:** Use the basic features of common digital tools to share content and collaborate demonstrating agreed behaviors, guided by trusted adults.

NZ:

- **CT P01:** They give these instructions, identify any errors in them as they are followed, and correct them (simple debugging).
- **CT P02:** Give, follow, and debug simple algorithms in computerized and non-computerised contexts.

GRADE 2 UNIT PLAN

CODE AVENGERS

Algorithms & Programming | **Fishful Thinking** | 10 hrs

Cross-Curricular Links

Math:

- Create effective strategies for performing addition, subtraction, multiplication, and counting techniques, and demonstrate the ability to communicate and explain these methods clearly to others.
- Measure and compare the area of two-dimensional shapes using non-standard units (e.g., counting unit squares).

English:

- Write informative/explanatory text in which they introduce a topic, use facts and definitions to develop points.
- Recognize and produce rhyming words. Use frequently occurring adjectives.

Science:

- Observe plants and animals in their natural environments.

#	Lesson Title	Lesson Description	Materials and Resources
1	Click To Start	Students will use Blockly, a visual programming language with drag and drop blocks, to create code. They will examine the freeplay scene and use different blocks from the toolbox. Students will follow instructions in a Scratch project to make a sea creature peg puppet. The lesson will conclude with a worksheet activity, allowing them to consolidate and reflect on their learning.	1. Unit slides 1-3 2. PR1 | L1 Using blocks to program 3. Peg puppets Scratch project 4. Building with Blockly worksheet 5. Clothespins - 1x for each student 6. Construction paper 7. Scratch workspace guide
2	Fin-tastic Fish	Students will engage in a variety of activities to learn about the different features and functions of fish. They will sequence blocks correctly to create images of sea creatures. An interactive experiment simulates fish gills in action. They will identify the different parts of a fish and use a graphic organizer to generate interesting facts about them.	1. Unit slides 4-6 2. Sea creature search (answers) support 3. PR1 | L2 Build sea creatures 4. 2x jars - filled with water 5. Rubber band 6. Coffee filters or paper towel 7. One scoop of coffee grounds or dirt 8. Fishy facts | A3 teacher version (answers)
3	Debugging Detectives	Students take turns in the roles of coders and debuggers. They work through a maze-making activity, and a coding activity that teaches them to break down programs into smaller blocks. The lesson encourages students to reflect on their experiences, make connections to real-world problem solving, and develop critical-thinking skills.	1. Unit slides 7-12 2. PR1 | L3 Add move blocks in sequence 3. Mini whiteboards (or paper) 4. Markers 5. Maze maker worksheet

Algorithms & Programming **Fishful Thinking** 10 hrs

4	Fishing For Teamwork	This lesson focuses on collaborative activities to help students learn important coding skills. Activities include identifying differences in images, debugging code, following origami instructions, and matching code blocks to sea creatures. Through these activities, students will learn to pay attention to detail, work efficiently with a partner, and develop problem-solving and critical-thinking skills.	1. Unit slides 13-14 2. Marine mismatch worksheet (answers) 3. PR1 | L4 Debug fishy creatures 4. Origami fish (easy) 5. Origami angel fish (harder) 6. Fixing fish activity (precut)
5	A Splash Of Color	In this lesson, students will play the game "Go fish!" to match patterns and colors. Afterwards, they will create their own fish design and write an explanation for their choice of colors and patterns. They will use a graphic organizer to justify design choices made and share their favorite feature.	1. Unit slides 15-16 2. Go fish! Cards 3. "Fish | Educational Video for Kids" video 4. PR1 | L5 A splash of color 5. Fish tales planning worksheet (example)
6	Ecosystem Events	Students will learn about food chains and their role in ecosystems. Through interactive activities, they will sequence events in a food chain and use various control blocks to create a program. This will help them understand how different living and nonliving things interact with each other in an ecosystem and the importance of maintaining a balanced food chain.	1. Unit slides 17-18 2. PR1 | L6 On click events 3. "What is a Food Chain?" video 4. The food chain game activity (cards)
7	Row By Row	Students will review the concept of area and learn how to calculate it by counting squares that fit inside a shape. They will then apply this concept by counting the number of squares that different animals take up and comparing their areas. Students will learn about grid references and how they help to position objects in programs.	1. Unit slides 19-22 2. PR1 | L7 Positioning on a grid 3. All squared up worksheet (answers) 4. Teacher example 5. Ocean I spy worksheet (answers)
8	Seuss Loops	Students watch a Dr. Seuss rap video and create their own rap using loops. They brainstorm rhyming phrases and work in groups to perform their raps. Students will identify different types of animation blocks, including the wait block, forever loop, and play sound block.	1. Unit slides 23-26 2. PR1 | L8 Animation with loops 3. Dr. Seuss Rap: "One Fish, Two Fish, Red Fish, Blue Fish" video 4. One fish, two fish rap strips
9	Putting It All Together	Students will explore creating animations in Scratch by examining an example project and remixing it. They will work in pairs to select sprites and apply animation blocks, add music, and record audio. They will experiment with adjusting the wait blocks to sync sounds and animations.	1. Unit slides 27-29 2. PR1 | L9 Storytelling 3. Group rap sheets from lesson 8 4. Microphones 5. One Fish, Two Fish Scratch (example)

Algorithms & Programming **Fishful Thinking** 10 hrs

10	Dive Into Decomposition	Students will watch a real-life application of decomposition where characters divide the problem of driving the car into smaller tasks. They will then apply this concept to solve five problems related to fish and the ocean, working in groups to rotate through problems. They will share and discuss strategies used.	1. Unit slides 30-32 2. PR0 | L10 Rock puzzles 3. Fluid thinking stations and worksheets 4. 4L container (clearly labeled with 4) 5. 7L container (clearly labeled with 7) 6. 1x large, 1x small clear container 7. Goldfish crackers 8. Counters, blank paper 9. "Toy Story Car Chase" video

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give it to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 3 UNIT PLAN

CODE AVENGERS

Algorithms & Programming | **Loopy Logic Lab** | 10 hrs

Unit Description

In this unit, students will explore algorithms, including comparing algorithms for the same task and breaking problems down. Students work through both **PR0: Code Crazy Creatures** and **CT2: Creature Feature Zoo** online courses. They'll learn about control structures such as conditional statements, loops, and variables for data storage and manipulation. Students will also explore different programming languages and their features, as well as get hands-on experience with coding projects. They will give precise instructions using directional language and use logical thinking tools and strategies.

Learning Outcomes

Target ages: US/CA Grade 3 | **AU** Year 3 | **NZ/UK** Year 4.

- Follow and write precise sequences of instructions that automate tasks.
- Use repeating patterns of instructions.
- Use events to initiate instructions.
- Break down complex tasks into simpler instructions, some of which can be broken down even further.

Curriculum Links

US:

- **1B-AP-08:** Compare and refine multiple algorithms for the same task and determine which is the most appropriate.
- **1B-AP-10:** Create programs that include sequences, events, loops, and conditionals.
- **1B-AP-11:** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
- **1B-AP-12:** Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
- **1B-AP-15:** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

AU:

- **AC9TDI4P02:** Follow and describe algorithms involving sequencing, comparison operators (branching), and iteration.

NZ:

- **CT P02:** Give, follow, and debug simple algorithms in computerized and non-computerised contexts.
- **CT P03:** Develop and debug simple programs that use inputs, outputs, sequence, and iteration.

UK:

- **KS2.CO2**: Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.
- **KS2.CO3:** Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Cross-Curricular Links

Math:

- Make sense of problems and persevere in solving them.
- Model with mathematics and use appropriate tools strategically.

English/Arts:

- Group ideas by recognizing similarities and differences.
- Recognise patterns in a range of media types.

GRADE 3 UNIT PLAN

CODE AVENGERS

Algorithms & Programming | **Loopy Logic Lab** | 10 hrs

#	Lesson Title	Lesson Description	Materials and Resources
1	Out Of Order	Students will draw on their past experience to discuss algorithms in real life and learn about a range of exciting programming blocks that they can drag into the editor to create crazy creatures. They learn that the order of the layers is important for the end result. Students also practice writing algorithms to sequence the steps required to draw objects.	1. Unit slides 1-5 2. PRO | T1: Introduction to creating programs 3. PRO | T2: Using blocks 4. Draw & order worksheet 5. Sorting sequences (A3, pre-cut) 6. "Why top universities teach drag and drop programming" video
2	Loopy Loops	Students identify where a series of actions are repeated within a piece of code. They explore how this helps programmers write more efficient code by using fewer blocks to create a path or repeat an action. This lesson encourages students to think multiplicatively (using multiplication) rather than additively (using repeated addition).	1. Unit slides 6-8 2. PRO | T3: Using on click events 3. PRO | T4: Using forever loops 4. PRO | T5: Create your own 5. Finding the part that repeats (answers) 6. Minimal Beat Machine website
3	Sum Up Solutions	Students will draw on their past experience of solving problems, and practice using strategies to prioritize decisions when constructing algorithms. They will identify potential solutions and any additional problems that might be associated with each solution.	1. Unit slides 9-11 2. CT2 | L1: Problem solving 3. CT2 | L2: What is an algorithm? 4. Got a problem with that? worksheet
4	Making Decisions	Students will learn about selection structures and how they control the flow of a program. The lesson begins with a demonstration that helps students understand simple conditions for making choices. They will use logic grids and flowcharts to choose between a range of outcomes.	1. Unit slides 12-15 2. Which Toy Story character are you? slides 3. Toy Story character flow chart image 4. CT2 | L3: Making decisions
5	On The Grid	Students explore a town map by moving one or more blocks north, south, east, or west. They specify their pathways' lengths in blocks, sometimes comparing the lengths or number of turns of two paths. Students will practice identifying a column and a row on a grid and give verbal instructions that use directions.	1. Unit slides 16-18 2. CT2 | L4: Position and orientation 3. CT2 | L5: Creating sequential algorithms - Tasks 1-4 4. Post-it notes

Algorithms & Programming **Loopy Logic Lab** 10 hrs

6	Point To Point	Finding the best way to get from one place to another on a network of roads is a problem that comes up often in everyday life. Students will practice planning routes to decide the route to take from one place to another. They will identify multiple paths and choose the best one. Studentswill then identify nodes and edges in computer science graphs.	1. Unit slides 19-21 2. CT2 | L5: Creating sequential algorithms - Task 5 3. CT2 | L6: Graphs and route planning 4. Pick-a-path worksheet 5. Count the squares worksheet 6. Computer science graphs (answers)
7	Puzzling Problems	This lesson is designed to equip students with an adaptable approach to solving a range of brain teaser problems, large or small. It includes a range of problems relating to various curriculum areas including English, Maths, Visual Arts and Language.	1. Unit slides 21-23 2. CT2 | L7: Logic puzzles 3. Logic relay worksheet (answers)
8	On One Condition	Students will recap conditions and how they can be used to control the flow of programs. They will evaluate conditions in statements and determine whether they are true or false. Students will then create their own conditions for a workout routine.	1. Unit slides 24-29 2. Large deck of cards or use Random Card Generator 3. CT2 | L8: Creating algorithms with decisions 4. Boolean Guess Who? (answers) 5. Coderobics worksheet
9	Abstraction	Students will learn about abstraction, decision making, and sorting. They use positional language and compass directions to describe how to return different beasties to their enclosures. Giving clear verbal instructions helps in writing algorithms and computer programs. Students also learn to identify unclear instructions.	1. Unit slides 30-31 2. "Abstraction - Computational Thinking" video 3. CT2 | L9: Using abstraction 4. Abstraction groups activity 5. Sorting morph animals (answers) 6. "Where have they gone?" cards and grids
10	Score & Explore	Students follow an algorithm to find the correct path, complete a picture puzzle, and find the next color in a pattern. They will conclude the story in CT2, consolidate learning with a Jeopardy quiz, and have an opportunity to explore programs in Scratch.	7. Unit slides 32-33 8. CT2 | L10: The escape 9. Funny flowcharts 10. Programming Jeopardy slides

Unit feedback/self-assessment survey links on next page

Algorithms & Programming **Loopy Logic Lab** 10 hrs

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 4 UNIT PLAN

CODE AVENGERS

Algorithms & Programming | **The Cycle Control Protocol** | 10 hrs

Unit Description

Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems. This unit introduces the concept of building programs with inputs and outputs using block-based coding. Students work through the **PR2: Gear Up For Safety** online course and use a variety of motion, sound, and event blocks to develop their own sequence-based programs. They will also investigate other fundamental programming concepts such as loops, repetition and iteration. Students will design games and remix or modify a variety of programs to solve user problems.

Learning Outcomes

Target ages: US/CA Grade 4 | **AU** Year 4 | **NZ/UK** Year 5.

- Use different types of blocks used to construct simple programs.
- Position sprites, backgrounds and movements using coordinates.
- Create programs with input and outputs.
- Explain the relationship between an event and an action.

Curriculum Links

US:

- **1B-AP-09:** Create programs that use variables to store and modify data.
- **1B-AP-10:** Create programs that include sequences, events, loops, and conditionals.
- **1B-AP-11:** Decompose problems into smaller, manageable subproblems to facilitate the program development process.
- **1B-AP-15:** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

AU:

- **AC9TDI4P02:** Follow and describe algorithms involving sequencing, comparison operators (branching), and iteration.
- **AC9TDI4P04:** Implement simple algorithms as visual programs involving control structures and inputand programs.

NZ:

- **CT P02:** Give, follow, and debug simple algorithms in computerized and non-computerised contexts.
- **CT P03:** Develop and debug simple programs that use inputs, outputs, sequence, and iteration.

UK:

- **KS2.C02:** Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.
- **KS2.C03:** Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms.

CA:

- Solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves:
  - **3-C3.1:** Sequential, concurrent, and repeating events.
  - **4-C3.1:** Nested events.
  - **5-C3.1:** Conditional statements and other control structures.

GRADE 4 UNIT PLAN

CODE AVENGERS

Algorithms & Programming | **The Cycle Control Protocol** | 10 hrs

Cross-Curricular Links

Health/PE:
Safety & Risk Management

- Meet and manage challenges and risks in positive, health-enhancing ways.
- Describe and demonstrate simple health care and safety procedures.

Technology:
Outcome Development & Investigation

- Investigate a context to develop ideas for feasible outcomes.
- Iterate through a concept-design process to develop a product.

English:

- Analyze the structure and content of procedural texts.
- Read and dissect informational texts and apply their learnings to a typed opinion piece of writing with supporting audio/visual displays.

Math:
Measurement & Geometry

- Give and follow instructions for movement that involve using, left, right, up, down, North, South, East and West, distances, directions, and half or quarter turns.
- Communicate, predict and record the results of translations, reflections, and rotations on plane shapes.
- Objects on a coordinate plane; Graph points on a coordinate plane.

#	Lesson Title	Lesson Description	Materials and Resources
1	Building Blocks Of Code	Students will explore algorithms in everyday contexts. They will use Scratch to program sequences in the theme of a bike pump track. Students will add sprites, backgrounds, and event blocks in Scratch's sandbox environment. They will then give one another feedback and show understanding of both the Code Avengers and Scratch coding environments.	1. Unit slides 1-4 2. "Bikes in Schools - Building a Pump Track" video 3. PR2 | L1: Sequential programs 4. Tick & check worksheet 5. Exit ticket worksheet (answers)
2	On The Grid	Students will begin by brainstorming ways to locate objects on a map. They will use X and Y references to mark numerous items on a grid, and to position themselves in relation to the origin point.	1. Unit slides 5-6 2. PR2 | L2: Positioning 3. Masking tape or painter's tape 4. Grid bingo 5. Track coordinate grid (caller) 6. Bike track grid (color) 7. Bike track grid (B&W) 8. Index cards or small squares of paper

Algorithms & Programming **The Cycle Control Protocol** 10 hrs

3	Making A Start	This lesson introduces the concept of events and actions. Students will identify a range of inputs and outputs, how they influence programs, and apply this understanding in a game of I/O Bingo. Students will explore the use of sensing, 'on key', 'on click', and control blocks. Then they'll combine event blocks to achieve a desired outcome.	1. Unit slides 7-10 2. PR2 | L3: Inputs & outputs 3. I/O Bingo activity 4. I/O Bingo player cards 5. I/O Bingo caller cards cards 6. Key events worksheet (answers) 7. Piano Scratch project
4	Looking Loopy	Students will identify parts of a program where repeat blocks can be used to make the program more efficient. They will explore the concept of loops in familiar examples and identify patterns within sequences. They will practice what they've learned in Scratch, adding repeat blocks and forever loops to make a sprite dance.	1. Unit slides 7-10 2. PR2 | L4: Loops 3. Finding the part that repeats worksheet (answers) 4. Duck dancing Scratch project 5. Dance loop Scratch project
5	Counting Clicks	In this skill-building lesson, students will use variables to track a value that changes over time, like a counter. They will use variables to store data that can be retrieved later and apply that learning to make a clicker game.	1. Unit slides 15-16 2. PR2 | L5: Interaction - review 3. Clicker game example Scratch project
6	Design Ahead	Students will use an iterative process to design according to user requirements. Like software developers, student teams research wearable technology (watching online videos and conducting online research), brainstorm a need that supports some aspect of human life, imagine their own unique designs, and sketch prototypes.	1. Unit slides 18-21 2. PR2 | L6: Helmet design 3. "Magpie swooping season" video 4. Hemet design worksheet 5. Assorted pieces of clothing
7	Re-Re-Remix	Students create a soundboard with buttons that trigger various sound samples using their developing programming skills. They will discuss the ethics around copying other people's work and use attribution for artists' music used in their programs.	1. Unit slides 27-28 2. PR2 | L7: Buttons 3. "Pop Culture (live mashup)" video 4. Soundboard Scratch project 5. Headphones recommended

Algorithms & Programming

The Cycle Control Protocol

10 hrs

8	Catching Bugs	Students will find the problems within an interactive program and fix them. They will use lists to offer a range of options to users. Students will then test a broken program to find code bugs and recognise this action as 'debugging'.	1. Unit slides 29-31 2. PR2 | L8: Lists 3. X,Y butterfly Scratch project 4. X,Y butterfly bugs worksheet (answers)
9	True Or False?	Students will make decisions about whether a number of statements are true or false. In the online lesson, they will group blocks together and show and hide questions to create a quiz. Students will also practice writing true and false questions about types of programming blocks in Kahoot!, and then test and provide feedback on each others' quizzes.	1. Unit slides 32-35 2. PR2 | L9: Quiz maker 3. Guess Who? cards (enough for 1x each) 4. How to Kahoot! instruction sheet 5. True/False Questions (offline version) 6. T/F Kahoot! template
10	Review	Students will reinforce vocabulary by making a mind map of concepts they have learned and playing charades, using a range of Scratch blocks and programming concepts. They will finish off the PR2 course and share their successes and failures in the programming tasks of this unit. Students share responses and feedback for improvement via Google Forms.	1. Unit slides 36-37 2. PR2 | L10: Revision 3. Vocabulary mind map 4. Programming charades cards 5. Student feedback form

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 4 UNIT PLAN

CODE AVENGERS

Programming Project **Code Your Own Adventure** 9 hrs

Unit Description

This programming project introduces students to conditionals and branching processes. Students will write a program that creates a story. In Scratch, they will use conditionals and branching techniques to make decisions and trigger actions. In the end, students will have created a pick-a-path story with multiple endings. They will test, debug, and refine their program. This project lets students explore and apply learning in a sandbox context over a series of lessons.

Learning Outcomes

Target ages: US/CA Grade 4 | **AU** Year 4 | **NZ/UK** Year 5.

- Write a branching story with multiple alternative endings.
- Decompose problems into step-by-step instructions to create an algorithm for a computer program.
- Develop a program using an iterative process involving design, implementation, and review.

Curriculum Links

US:

- **1B-AP-10:** Create programs that include sequences, events, loops, and conditionals.
- **1B-AP-11:** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
- **1B-AP-13:** Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
- **1B-AP-14:** Observe intellectual property rights and give appropriate attribution when creating or remixing programs.
- **1B-AP-16:** Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.
- **1B-IC-18:** Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.
- **1B-IC-19:** Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.
- **1B-AP-15:** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

AU:

- **ACTDI019:** Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition).
- **ACTDI020:** Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input.
- **ACTDIP026:** Design the user experience of a digital system, generating, evaluating and communicating alternative designs.

NZ:

- **CT P02:** Give, follow, and debug simple algorithms in computerized and non-computerised contexts.
- **CT P03:** Develop and debug simple programs that use inputs, outputs, sequence, and iteration.

UK:

- **KS2.2:** Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.
- **KS2.1:** Solve problems by decomposing them into smaller parts.
- **KS3:** Create, reuse, revise, repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability.

CA:

- **3-C3.1:** Sequential, concurrent, and repeating events.
- **4-C3.1:** Nested events.
- **5-C3.1:** Conditional statements and other control structures.

GRADE 4 UNIT PLAN

CODE AVENGERS

Programming Project | **Code Your Own Adventure** | 9 hrs

Cross-Curricular Links

Writing:
Text Types & Purposes

- Use effective technique, well-chosen details, and well-structured event sequences to write narratives about actual or imagined events.

Production & Distribution of Writing

- Plan, revise, edit, rewrite, and attempt new approaches to improve writing.
- Use technology, notably the Internet, to write, publish, and collaborate with others.
- Present ideas to a group and establishing and using rules/guidelines for their review processes/presentations, posing follow up questions.
- Make connections while providing feedback to a peer.
- Make connections between text of a story and visual or oral presentation of text.

#	Lesson Title	Lesson Description	Materials and Resources
1	Evolution of Storytelling	Students will investigate and discuss a range of innovations that technology has adapted or improved over time. They will explore how the concept of storytelling has allowed us to interact and tell stories in a new way; with linear and non-linear structures. Ultimately, students should come to understand how innovation can drive the development of new products and services, and explore ways to use it to create their own original ideas.	1. Unit slides 2-10 2. "The playful wonderland behind great inventions" video 3. Tarmac or concrete courts and chalk
2	Hook, Line, and Sinker	This lesson is designed to teach students the fundamentals of writing story hooks. By the end of the lesson, students will have identified multiple story hooks and tried their hand at writing their own. Students will be given a random object and are encouraged to create their own story hooks for each. Finally, students can share their story hooks with the class and get feedback.	1. Unit slides 11-17 2. Sorting story starters worksheet (answers) 3. Devices encyclopedia entries 4. Brown paper bags with assorted objects 5. Sticky notes
3	Audience and Purpose	Students will be tasked with analyzing the audiences and the purpose for their writing. They will develop a plan for their story's characters, setting, problem and solution, and then collaborate by swapping stories with their peers. The lesson will conclude with a discussion on the impact of building upon the work of others, and reflections on the collaborative experience.	1. Unit slides 18-22 2. Pass it on worksheet

Programming Project **Code Your Own Adventure** 9 hrs

4	Planning a Project	This lesson begins with an example of the possibilities within Scratch to create a pick-a-path adventure. Students will explore the project parameters and assign roles within their group.	1. Unit slides 23-25 2. "You vs. Wild | Interactive Series ft. Bear Grylls" video 3. Pick-a-Path Project Workbook p1-4
5	Story Planning	Students will recognize and understand common story structures.They will use graphic organizers to sequence the main points of the stories. They will share their story plan and provide constructive feedback while also recognising story elements in examples and the story plans of others.	1. Unit slides 28-30 2. Pick-a-Path Project Workbook p5 3. Beginning, middle, end cards 4. Story planning checklist 5. "Coin Operated" video
6	Split Ends	Students start with a warm up activity to get thinking about alternative endings to stories and how they can apply this to commonly known stories. Students will be given an offline flowchart graphic organizer to develop their stories with multiple choices and endings. The lesson concludes with students marking success criteria for their story planning	1. Unit slides 31-32 2. "Fixed Fairy Tales" video 3. Split ends worksheet 4. Pick-a-Path Project Workbook p5-6 5. Flowchart Planning Exemplar 6. Flowchart Planning (support) 7. Camping Stinks - Scratch exemplar
7	Turning Text Into Code	Students build their pick-a-path story in Scratch. They will break down their story into key decisions made by a user and adapt a plan as time limitations or resource restraints impact their project. Students will run and test their code as they build their programs as an iterative design method.	1. Unit slides 33-37 2. Pick-a-Path Project Workbook p6-10 3. Camping Stinks - Scratch exemplar 4. Pick-a-Path template (support)
8	Testing & Debugging	Students will consider what steps should be taken to make their story more engaging. They will test one anothers' programs, use a bug map to record their feedback and will reflect on the skills they have learned.	1. Unit slides 38-40 2. Pick-a-Path Project Workbook p12 3. Bug Map worksheet 4. "Want to be a Videogame Tester?" video
9	Finishing Touches	In this closing lesson students will polish and refine their programs. They will share their projects in a walkaround gallery and reflect on the skills they have learned and the success of their project.	1. Unit slides 38-41 2. Pick-a-Path Project Workbook p12

Unit feedback/self-assessment survey links on next page

Programming Project **Code Your Own Adventure** 9 hrs

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 5 UNIT PLAN

CODE AVENGERS

Algorithms & Programming | **Castle Coding Capers** | 8 hrs

Unit Description

Students work through the **PR3: Larsson Castle Mystery** online course. Through this they will learn a range of programming structures which will enable them to develop digital artifacts including animations, games, and shopping apps. Lessons will also include offline activities, class discussions, quizzes, and freeplay time in Scratch to practice skills.

Learning Outcomes

Target ages: US/CA Grade 5 | **AU** Year 5 | **NZ/UK** Year 6.

- Create programs that include sequences, inputs, loops, conditionals and variables.
- Follow algorithms to identify bugs in programs.
- Identify websites to source free images.
- Research and inspect other's code that could be modified or remixed.

Curriculum Links

US:

- **1B-AP-09:** Create programs that use variables to store and modify data.
- **1B-AP-10:** Create programs that include sequences, events, loops, and conditionals.
- **1B-AP-12:** Modify, remix, or incorporate portions of an existing program into one's own work.
- **1B-AP-14:** Observe intellectual property rights and give appropriate attribution.
- **1B-AP-15:** Test and debug (identify and fix errors) a program or algorithm.
- **1B-AP-17:** Describe choices made during program development using code comments, presentations, and demonstrations.

AU:

- **AC9TDI6P05** Implement algorithms as visual programs involving control structures, variables and input.
- **AC9TDI6P02** Design algorithms involving multiple alternatives (branching) and iteration.

NZ:

- **CT P03**: Develop and debug simple programs that use inputs, outputs, sequence, and iteration.
- **CT P04:** Create programs that use inputs, outputs, sequence, basic selection using comparative operators, and iteration.
- **DD P01:** Develop, manipulate, store, retrieve, and share digital content in order to meet technological challenges.

UK:

- **KS2:** Solve problems by decomposing them into smaller parts.
- **KS2**: -Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.

GRADE 5 UNIT PLAN

CODE AVENGERS

Algorithms & Programming **Castle Coding Capers** 8 hrs

Cross Curricular

Math:

- Programs require students to position and move sprites. This process can help reinforce Math concepts such as coordinates, cardinal directions and angles.
- Programming involves using variables and operations to manipulate data. This process is similar to Math, where students use variables and operations to solve equations and perform mathematical operations.

English:

- Group or pair activities help students develop their speaking and listening skills.
- Writing a program requires students to think logically and sequentially about the steps involved in solving a problem. This skill is transferable to writing where students need to think logically about the structure of their writing, such as the order of paragraphs and the flow of ideas.

#	Lesson Title	Lesson Description	Materials and Resources
1	Out Of Order	Students position and rotate sprites, using x and y coordinates and angle degrees . They also adjust the size of shapes, using width and height values. They learn that computer programs run from top to bottom and the order of the blocks is important to achieve the desired effect for the shield design.	1. Unit slides 2 - 8 2. PR3 | L1: Shield design - Layers 3. Shield design activity 4. Example medieval shields 5. Shield templates: Medieval, Round, Oval 6. Pencil, pen, ruler 7. Paper, colored pencils (or felt tips, markers, or paints), glue
2	Algorithms & Attribution	Students create simple animations to retell the battles of Lord Lucas and Emperor Gustav . They use the glide, hide and wait blocks. They follow algorithms, detect code block bugs, and recognize this action as 'debugging'. The concept of observing intellectual property rights is introduced by visiting unsplash and freepik sites.	1. Unit slides 9 - 15 2. PR3 | L2: Animated Story - Sequence 3. Medieval shield examples 4. Student's shield designs from lesson 1. 5. Unsplash 6. Freepik 7. Google Advanced Image Search

Algorithms & Programming

Castle Coding Capers

8 hrs

3	Iterate, Animate	Students use repeat blocks to determine the number of iterations and allow sprites to be animated. They are given a series of animation challenges using the Code Avengers course and the opportunity to create their own animations in Scratch.	1. Unit slides 16 - 18 2. PR3 | L3: Animated Story - Loops 3. Bear Scratch animation 4. Bear Scratch animation (extended) 5. Setting up a Scratch account.
4	Valiant Variables	Students learn that to store information (text or numbers) in a computer program they will need to use a variable. They set up a score variable called coins variable for their 'Whack a ghoul' game. They use an on-click event blocks to detect when there has been a successful whack.	1. Unit slides 19 - 26 2. Snake game 3. Variables video 4. PR3 | L4: Whack-a-Ghoul - Variables 5. Pack of Cards (Remove J, Q, K, Joker) 6. Counting Cards recording sheet 7. Whiteboard 8. Variables Kahoot!
5	Conditions & Comparatives	Students learn about conditional loops, which only run when a condition is true. First they look at conditions that use operators and identify whether they are true or false. Students further develop their whack a ghoul game using a while loop to control the game length. They also create an algorithm which includes a repeat until loop.	1. Unit slides 27 - 32 2. Operator worksheets (Answers) 3. PR3 | L5: Whack-A-Ghoul - While loops 4. Pool plunge algorithm template 5. Pool plunge Scratch project 6. Operator exit pass
6	Map App	Students develop an interactive castle map by using compass sprites (N, E, S, W) that play informational sounds when clicked. A variable block stores the hall name. They use a forever block to keep checking the hall variable and the appropriate hall background sprite is displayed depending on the value.	1. Unit slides 33 - 38 2. "The Directions Song" video 3. PR3 | L6: Castle Map - Play sounds 4. PR3 | L7: Castle Map - Forever if 5. Random wallpaper Scratch project
7	Mastering The Maze	Students develop a maze game about Lord Lucas escaping to Sweden by combining loops, variables, if statements, and sensing events (on key, on touch). New blocks called 'list' and 'next on list' are introduced.	1. Unit slides 39 - 42 2. Castle maze (Solution) 3. PR3 | L8: Maze Game - Lists & Keys 4. PR3 | L9: Maze Game - On Touch 5. Pong game cards

Algorithms & Programming **Castle Coding Capers** 8 hrs

8	Verify The Vocab	Students start the lesson by reviewing the unit vocabulary. Then they make a castle gift shop by positioning sprites, using events, and variables to update the total spend. While loops are used to ensure a free gift when 30 coins are spent. They finish by completing a self paced kahoot to show their understanding of the unit.	1. Unit slides 43 - 48 2. Vocabulary word search (Answers) 3. PR3 | L9: Maze Game - On Touch 4. PR3 | L10: Gift Shop - Review 5. Pong game cards 6. Unit vocabulary sheet 7. End of unit Kahoot! 8. End of unit feedback survey 9. End of unit self assessment: a. Google form b. Printed PDF

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 6 LESSON PLAN

CODE AVENGERS

Algorithms & Programming | **The Appetite Algorithm** | 10 hrs

Unit Description

Students work through the **CT3: Saving Food Avengers** online course. They will gain a basic understanding of what algorithms are, in both their everyday lives and computer science. Students will learn how to decompose problems to be able to write algorithms that can be "run" and "debugged" by their peers. They will use loops and conditions to optimize their algorithms and make their code easier to read.

Learning Outcomes

Target ages: US/CA Grade 6 | **AU** Year 6 | **NZ/UK** Year 7

- Explain what an algorithm is and how they relate to real life and computer programs.
- Design, write and debug programs that accomplish specific goals.

Curriculum Links

US:

- **2-AP-16:** Incorporate existing code, media, and libraries into original programs, and give attribution.
- **2-AP-13:** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

AU:

- **ACSTDI8P05:** Design algorithms involving nested control structures and represent them using flowcharts and pseudocode.
- **ACSTDI6P02:** Design algorithms involving multiple alternatives (branching) and iteration.

NZ:

- **CT P03:** Students decompose problems into step-by-step instructions to create algorithms for computer programs. They develop and debug simple programs that use inputs, outputs, sequence and iteration (repeating part of the algorithm with a loop).

UK:

- **KS3:** Use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; design and develop modular programs that use procedures or functions.

GRADE 6 LESSON PLAN

CODE AVENGERS

Algorithms & Programming | **The Appetite Algorithm** | 10 hrs

Cross-Curricular Links

Math:
Number & Algebra

- Sequential number/spatial patterns: find rules for the next member, connecting with their ordinal positions, using tables, graphs, and diagrams to find relationships.

Measurement & Geometry

- Give and follow instructions for movement that involve using left, right, up, down, North, South, East, West, distances, directions, and half or quarter turns.
- Communicate, predict and record the results of translations, reflections, and rotations on plane shapes.
- Objects and graphing points on a coordinate plane.

Technology:
Technological Modeling

- Investigate different forms of functional modeling that are used to explore possibilities and to justify decision making.
- Prototyping can be used to justify refinement of outcomes.

Outcome Development & Investigation

- Investigate a context to develop ideas for feasible outcomes

#	Lesson Title	Lesson Description	Materials and Resources
1	What Is An Algorithm?	In computer science, we build tools to help us solve problems relating to games, data, and other computer programs. This lesson introduces the concept of an "algorithm" as a set of instructions used to solve a problem and establishes the context for our later exploration of building and adapting algorithms.	1. Unit slides 2-8 2. CT3 | L1: Sequential algorithms 3. Paint by coordinates worksheet 4. "Inside Ocado's Distribution Warehouse" video
2	Unplug & Debug	In this lesson students use various tactics to identify and locate bugs in a range of algorithms. They experience why clear instructions are important and use a “debugging mindset” to solve problems.	1. Unit slides 9-12 2. CT3 | L2: Build & debug programs 3. Lego bricks, building bricks, multilink cubes 4. Twinkle twinkle little star Scratch project 5. Debugging: spot the difference worksheet (answers) 6. Lightbot [browser or installed on device]

Algorithms & Programming **The Appetite Algorithm** 10 hrs

3	Decomposition	The students will deconstruct simple sequences into their smallest steps. They will develop skills in spatial awareness and their ability to break a shape down into basic parts by working through guillotine problems. The process of decomposing becomes less complicated as the shape is disassembled into smaller and smaller pieces.	1. Unit slides 13-17 2. "The Cup Song" video 3. Plastic cups, enough for one each 4. CT3 | L3: Algorithms to solve math problems 5. Guillotine problem worksheet 6. **Extension:** CT3 | L4: Logic & problem solving
4	Perfect Patterns	By rotating regular shapes around a circle at specific angles, students will create iterative patterns. They will then investigate patterns created by spinning pictures at various angles, determining how many times they must perform an action in order for it to turn a full 360 degrees.	1. Unit slides 18-21 2. CT3 | L5: Iteration & patterns 3. Angles and regular shapes worksheet 4. Repeating Pictures Scratch project
5	Optimizing Algorithms	Students will use logical thinking to predict the behavior of programs, and understand that there can be more than one algorithm for the same problem. Students will read and draw flowcharts to make more complex sequences.	1. Unit slides 22-26 2. Figuring out flowcharts worksheet (answers) 3. CT3 | L6: Optimizing algorithms 4. "What is the Ant Colony Optimization Algorithm?" video 5. Flowchart match activity 6. **Extension:** CT3 | L7: Ciphers
6	Decisions & Conditions	This lesson introduces the concept of selection when building conditional algorithms through a variety of burgers. Conditional algorithms have decision points that determine whether the algorithm branches and this impacts the final output.	7. Unit slides 27-29 8. CT3 | L8: Conditional algorithms - Tasks 1-4 9. Ingredients cards 10. Burger algorithm instructions 11. Burger design worksheet
7	Comparatives	Students will use a logic grid to decide which customers to serve 4 different burgers. They explore the concept of more complex conditionals, using comparative operators to define the outcomes of their quizzes.	1. Unit slides 30-32 2. CT3 | L8: Conditional algorithms - Tasks 4-5 3. Ingredients cards 4. Blank Logic Puzzle worksheet 5. Logic Puzzle worksheet 6. "Coding Basics: if Statements" video 7. **Extension:** CT3 | L9: Finding patterns

Algorithms & Programming **The Appetite Algorithm** 10 hrs

8	Show You Know	This lesson is an opportunity for students to review key concepts and vocabulary, assess their learning, and consolidate learning in core algorithmic concepts.	1. Unit slides 33-35 2. CT3 | L10: Revision & the party day 3. Programming and algorithms review quiz
9	Proportionate Coordinates	This lesson allows students to practice using block- based programming and compare it to JavaScript programming. They will use both screen and Cartesian Coordinates to draw a range of shapes. Students will build increasingly complex flags using block functions.	1. Unit slides 36-39 2. JS101 | Sequence Grid Challenge 3. JS123 | Blockly 4. **Extension**: JS112 | Flags Challenge 5. Whiteboards (if available) 6. Cartesian coordinates Scratch project 7. Battleships coordinate game website
10	Intro To Python	Students will be introduced to a new programming language, Python. They write simple sequential programs to move a turtle around a drawing canvas.	1. Unit slides 40-42 2. PY100 | Intro to Turtle Graphics 3. "Awesome Designs with Python Turtle" video 4. Mentimeter reflection tool & presentation link

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 6 UNIT PLAN

CODE AVENGERS

Data & Analysis

The Information Transformation

9 hrs

Unit Description

Computers display data (including decimal numbers, words, images and videos) using only the digits 0 and 1. Students learn about binary numbers and how to convert them into decimals. The majority of the unit focuses on numerical data, how it is collected, analyzed, formatted, and presented in data visualizations.

Students are provided with opportunities to learn skills through the DR200: Data Rep DJ and DA60: Your Pizza Is Surv...eyed courses on the Code Avengers platform. Alongside these, they complete a series of practical exercises to reinforce the skills. Google Sheets is used as the spreadsheet application but the unit could be adapted for use with Microsoft Excel.

Learning Outcomes

Target ages: US/CA Grade 6 | **AU** Year 6 | **NZ/UK** Year 7

- Identify a range of data types that can be represented using binary.
- Analyze numerical data using sorting, filtering, formulas, and functions.
- Use data visualizations to display information.
- Apply a range of formatting techniques to improve design and usability of data.

Curriculum Links

US:

- **2-DA-07** Represent data using multiple encoding schemes.
- **2-DA-08** Collect data using computational tools and transform the data to make it more useful and reliable.
- **2-IC-22:** Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

AU:

- **AC9TDI6K04** Explore how data can be represented by off and on states (zeros and ones in binary).
- **AC9TDI6K03** Explain how digital systems represent all data using numbers.

NZ:

- **CTDT P03:** They understand that digital devices store data using just two states represented by binary digits (bits).
- **DDDO P02:** They can select from an increasing range of applications and file types to develop outcomes for particular purposes.

UK:

- **KS3 Computing:** Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analyzing data and meeting the needs of known users.

GRADE 6 UNIT PLAN

CODE AVENGERS

Data & Analysis | **The Information Transformation** | 9 hrs

Cross-Curricular Links

The skills learnt through this unit can be put into practice and built upon in most areas of the curriculum. Teachers need to choose a topic to allow students to gather some authentic data which can be analyzed to draw conclusions or make predictions.

Math:
There is a strong focus on statistics including data collection (surveys) and data displays (dot plots, pie charts, bar graphs).

- **UK KS3 Mathematics:**
  - Understand and use place value for decimals, measures, and integers of any size.
  - Define percentage as 'number of parts per hundred'.
  - Construct and interpret appropriate tables, charts, and diagrams.

- **AU Mathematics:**
  - ACMSP147: Interpret and compare a range of data displays, including side-by-side column graphs for two categorical variables.
  - ACMNA123: Select and apply efficient mental and written strategies and appropriate digital technologies to solve problems involving all four operations with whole numbers.
  - ACMNA131: Make connections between equivalent fractions, decimals and percentages.

- **NZ Mathematics:**
  - NA4-3 | NA4-5 | S4-1

#	Lesson Title	Lesson Description	Materials and Resources
1	What Is Binary?	Students discuss the difference between quantitative and qualitative data, and learn how computers store both types of data as numbers, using the binary number system. They will model the binary number system through a physical activity.	1. Unit slides 2 - 21 2. "What are binary numbers" video 3. "How Binary numbers work" video 4. Binary cards (Print 5 sets back to back) 5. DR200 | T1 What is data representation? 6. Binary boxes activity 7. Yellow coloring pencil 8. Calculator

Data & Analysis

The Information Transformation

9 hrs

2	Bits Of Color	Students explore how basic binary is used to represent individual pixels, which can be black or white. They progress to inspecting larger images made up of pixels of differing colors which have their own unique RGB code which can be represented in binary or hexadecimal.	1. Unit slides 22 - 27 2. DR200 | T2 Represent images in binary 3. Pixel Paint worksheet (answers) 4. Pixel Viewer 5. Decimal to binary converter* 6. Color representation worksheet 7. RGB Mixer 8. Pixel Perfect Kahoot
3	Simple Surveys	Traditional and modern ways of collecting data with a focus on surveys are explored. Issues such as question design, bias and privacy are also touched on. Students collect data via a manual survey, organizing the data into a table and then creating a paper based column chart to analyze the results.	1. Unit slides 28 - 29 2. "World of data" video 3. DA60 | L1: Collect Data 4. Survey Template Table Template 5. Column Chart Template 6. Pens or Marker to add answers 7. 4 Colored pencils for column chart 8. Analysis Spreadsheet (xlsx version)
4	Ships & Sheets	Students explore the physical environment and identify keywords, to become familiar with spreadsheet applications. They practice cell references through a fun battleship game. They also consider the purpose of a spreadsheet and explore different situations in which they may be used.	1. Unit slides 30 - 35 2. Spreadsheets video 3. DA60 | L2: Enter & store data 4. "How to play: paper/pen battleships" video 5. Battleships - Whole Class 6. Battleships - 2 Player
5	Sort & Filter	This lesson focuses on using a spreadsheet as a database to store, organize, and locate data. Students will learn to sort data and use filters to find relevant information.	1. Unit slides 36 - 42 2. Paper and pens for sorting activity 3. DA60 | L3: Analyze & Visualize Data - Tasks 1-2 4. Filtering Database (xlsx version) 5. Filtering Superheros (answers) 6. Sorting Spreadsheet (xlsx version)

Data & Analysis

The Information Transformation

9 hrs

6	Formulas & Function	Students gain hands-on opportunities to create a range of formulas and functions. There is a strong focus on the mathematical understanding of the center of data sets with a practical activity to support.	1. Unit slides 43 - 49 2. Students in groups of 5 or 6 of mixed math ability 3. Measuring tapes /Pens 4. Calculator 5. Height measure 6. DA60 | L3: Analyze & Visualize Data - Tasks 3-4 7. Formula/Function Practice (xlsx version) 8. Kahoot!
7	Visualize Data	Students are exposed to a range of data visualization approaches. They will explore a number of different data sets on the "Our World in Data" website. At the end of the lesson they should have a good understanding of when to use different data visualization techniques.	1. Unit slides 50 - 53 2. DA60 | L3: Analyze & Visualize Data - Task 5 3. 'Choose-the-right-chart" article 4. Our world in data 5. Data visualizations worksheet (answers) 6. Extension videos "Using design techniques" and "Data Representation & Misrepresentation"
8	Fantastic Formatting	Students complete a Code Avengers lesson to learn about a range of different formatting techniques. They apply these new skills through a practical exercise of reproducing a well designed timesheet.	1. Unit slides 54 - 46 2. Readable fonts article 3. DA60 | L4: Format data 4. Formatting practice 5. Weekly Timesheet pdf 6. Unformatted timesheet for students (xlsx file)
9	End Of Unit Assessment	The students watch two videos to learn how to share their spreadsheets with classmates and how to add comments. There is also a short discussion on how to give feedback effectively. The majority of the lesson is spent on an assessment, analyzing and formatting a basketball scoring spreadsheet. There is an opportunity towards the end to share and comment on other's work.	1. Unit slides 57 - 58 2. Students in groups of 3/ 4 for feedback 3. Sharing documents video 4. Notes & comments video 5. Basketball scoring rules article 6. Assessment instructions 7. DA60 | L5: Final output 8. CA Coders spreadsheet (xlsx version) (exemplar)

Unit feedback/self-assessment survey links on next page

Data & Analysis

The Information Transformation

9 hrs

Unit Survey

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 6 LESSON PLAN

CODE AVENGERS

Networks & Security

The Authentication Investigation

10 hrs

Unit Description

Students work through the **NS600: Operation Cloud** online course. They will explore elements of networks including physical hardware and data centers as well as the security measures taken by people who work in the technology industry as cyber security professionals. This unit covers some cyber safety concepts such as authentication processes, password hygiene, and social engineering.

The learning in this unit culminates in a final project where students plan, prototype, and build a foolproof security system to protect their stuff. They will use the design process to define the problem, generate ideas, prototype, test, build and analyze their solution. Students will use critical thinking skills to evaluate evidence, assess thinking, formulate strategies, set criteria, make decisions, and gather data.

Learning Outcomes

Target ages: US/CA Grade 6 | **AU** Year 6 | **NZ/UK** Year 7

- Compare cloud computing with older data storage methods and recall facts about data centers.
- Describe the methods of authentication and outline some of the risks attached to each method.
- Apply various strategies and tools used in security systems to minimize risks.

Prior Learning:

- Students should know the difference between qualitative and quantitative data.

Curriculum Links

US:

- **2-NI-05:** Explain how physical and digital security measures protect electronic information.
- **1B-IC-18:** Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.
- **2-IC-21:** Use unique passphrases and explain the risks of password re-use.
- **1B-AP-13:** Use an iterative process to plan the development of a project by including others' perspectives.

AU:

- **AC9TDI8P13:** Explain how multi-factor authentication protects an account when the password is compromised and identify phishing and malware threats.

NZ:

- **DO P03:** They understand that with storing data comes responsibility for ensuring security and privacy.

UK:

- **KS3:** Understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns.

GRADE 6 LESSON PLAN

CODE AVENGERS

Networks & Security

The Authentication Investigation

10 hrs

Cross-Curricular Links

Science:
Nature of science & physical world
- Use a growing knowledge of science when considering issues of concern to them and identify and describe everyday examples of sources of energy, forms of energy, and energy transformations.

Math:
Measurement
- Use linear scales and metric units for length, area, volume and capacity, weight (mass), angle, temperature, and time.
- **NZ Mathematics:** Geometry & measurement GM3-1 | S3-1 | S3-2.

Technology:
Technological Modeling
- Investigate different forms of functional modeling that are used to explore possibilities and to justify decision making.
- Prototyping can be used to justify refinement of outcomes.
- Investigate a context to develop ideas for feasible outcomes.
- **NZ Technology:** TM-L3.

#	Lesson Title	Lesson Description	Materials and Resources
1	Networks	Students learn about the origins of the internet and how it has grown into a network of networks. They explore the rise of cloud computing which allows data to be accessed anywhere, anytime on multiple devices.	1. Unit slides 3-6 2. "Growth of the internet" video 3. "How does the internet work?" video 4. Internet Kahoot! 5. NS600 | Task 1: History of digital audio storage 6. NS600 | Task 2: What is the cloud? 7. Back up worksheet (answers)
2	Data Centers	Students explore physical data centers and learn about what measures are in place to keep information secure. This lesson focuses on the physical infrastructure of data storage and management where the location of data centers is often dependent on environmental factors such as natural disaster hazards and green energy sources.	1. Unit slides 7-10 2. NS600 | Task 3: Data centers 3. NS600 | Task 4: Security measures 4. Data center scavenger hunt worksheet (answers) 5. Find your closest data center website 6. We live in the cloud 3D tour website

Networks & Security

The Authentication Investigation

10 hrs

3	Cybersecurity Experts	Students will investigate different types of people who are involved in cybersecurity. They explore the concept of hacking as well as the various types of hackers and strong and weak points in a network.	1. Unit slides 12-15 2. Hack attack activity 3. NS600 | Task 5: Careers 4. IC300 | Task 1: What is a hacker?
4	Authentication	This lesson provides hands-on opportunities to understand how computers authenticate users. There are opportunities for discussion with students around the ethical ramifications, advantages, and disadvantages of authentication methods such as facial recognition.	1. Unit slides 17-18 2. IC300 | Task 2: What is authentication? 3. IC300 | Task 3: Password hygiene 4. Facial recognition activity 5. Faces A & Faces B worksheets 6. Tracing paper (5x A4)
5	Social Engineering	This lesson encourages students to consider how we verify identity and truthfulness of online information and how those verification processes can be manipulated by someone attempting to run a scam. There is a strong emphasis on human error and how it can compromise a security system.	1. Unit slides 20-24 2. IC300 | Task 4: Social Engineering 3. Catching phish worksheet (answers) 4. Phishing for learning exit ticket (form)
6	Project Ignition	This lesson introduces the Home Alone design project. There is an opportunity to co-construct a marking rubric, and unpack learning outcomes. Students begin the design process by defining the problem in the terms outlined within the project booklet. They work through the ideation process to generate concepts.	1. Unit slides 25-28 2. Home alone project workbook (p. 1-4) 3. "Explosion-proof fire fighting robot" video
7	Ideate	Students brainstorm ideas for their project and begin to make plans.	1. Unit slides 29-31 2. Home alone project workbook (p. 5) 3. Networks & security Kahoot 4. Security periodic table
8	Prototype	Students use the ideas they generated in the last lesson to plan and prototype their security systems based on a blueprint of their homes. This lesson focuses on the testing process that occurs during the prototyping phase of design.	1. Unit slides 32-32 2. Home alone project workbook (p. 6-7) 3. Prototype planning worksheet 4. "Explosion-proof fire fighting robot" video 5. Math/building materials e.g. base ten, multilink cubes

Networks & Security

The Authentication Investigation

10 hrs

9	Build	This lesson is largely student-led as groups choose an appropriate tool and build their security models either physically or digitally.	1. Unit slides 34-35 2. Home alone project workbook (p. 8)
10	Evaluate	In this lesson, students are shown how to build and distribute a google forms survey. They construct questions to evaluate their security system designs and use the information gathered to write a short explanatory paragraph about their security system.	1. Unit slides 36-38 2. Savvy surveys form 3. Home alone project workbook (p. 9) 4. Survey template form

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 7 UNIT OVERVIEW

CODE AVENGERS

Algorithms & Programming | **Scripting Showdown** | 10 hrs

Unit Description

This unit exposes students to two programming languages, JavaScript and Python. Students learn how to create programs that accept inputs by using variables. They use operators to construct conditional statements to enable the program to make selections. They create programs in a number of different contexts including a simple game, drawing designs, quizzes and Math problems. Code Avengers courses used are: JS0: Intro to Javascript, JS102: Intro to Graphics, JS103: Drawing Flags, JS105: Food Frenzy, PY0: Intro to Python, PY101: Grid Challenge.

Learning Outcomes

Target ages: US/CA Grade 7 | **AU** Year 7 | **NZ/UK** Year 8

- Create programs with conditional and iterative structures, using two different text- based programming languages.
- Create programs that accept input numbers, to process mathematical problems and output a solution.
- Analyze code to identify bugs.

Curriculum Links

US:

- **2-AP-11:** Create clearly named variables that represent different data types and perform operations on their values.
- **2-AP-13:** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- **2-AP-12:** Design and iteratively develop programs that combine control structures.
- **2-AP-14:** Create procedures with parameters to organize code and make it easier to reuse.
- **2-AP-19:** Document programs in order to make them easier to follow, test, and debug.

AU:

- **AC9TDI8P09:** Implement, modify and debug programs involving control structures and functions.

NZ:

- **CT P04:** Creating programs that use inputs, outputs, sequence, basic selection using comparative operators, and iteration.

UK:

- **KS3:** Use 2 or more programming languages, (1+ is textual); make appropriate use of data structures (lists, tables, or arrays).

CA:

- **7-C3.2:** Read and alter existing code that involves events influenced by a defined count and/or subprogram.

Cross-Curricular Links

Math:

- Geometry > Area > Triangles, rectangles and circles.
- Geometry > Compass directions, distances, and grid references.
- Number > Operations with decimal.

English:

- Writing a program requires attention to detail to ensure that code is written correctly and that there are no syntax errors.
- Writing code comments helps students practice technical writing skills and learn how to communicate their ideas clearly and concisely.

GRADE 7 UNIT OVERVIEW

CODE AVENGERS

Algorithms & Programming **Scripting Showdown** 10 hrs

#	Lesson Title	Lesson Description	Materials and Resources
1	Tangrams	Students are introduced to screen coordinates. They also learn about tangrams and practice making different designs. They combine these skills to write algorithms using coordinates that other people can follow to create shape designs.	1. Unit slides 2 - 9 2. JS0 | L1: Coordinates Grid 3. JS0 | L2: Commands Grid 4. Javascript 0 Progress 5. "Story of the Tangram Square" video 6. Tangram Squares - Color (B&W) 7. Tangram Worksheet (Answers) 8. JS102 | Intro to Graphics 9. Tangram algorithm sheet 10. Ruler & pencil
2	Variables	This lesson introduces the concept of a variable through movement on a grid. Students also consider the importance of computer programs needing to make decisions e.g. an automated cleaner like a Roomba.	1. Unit slides 10 - 19 2. Variables video 3. "Roomba i7 overview" video 4. JS0 | L5: Variable grid 5. JS0 | L7: Conditionals revision 6. JS0 | L9: Variables (extension) 7. JS0 Teaching guide 8. Exit passes (2 per page) 9. Balls or bean bags for catching activity
3	Input | Process | Output	Students start by programming some simple commands in JavaScript to take user input, and output some data. These basic commands are combined with IF statements, expanding the capability of the program to check for conditions and make decisions. The outcome of the lesson is to create the foundations for designing a quiz.	1. Unit slides 20 - 26 2. "CS Basics :Input Process Output" video 3. JS0 | L3: Alerts 4. JS0 | L6: Prompts 5. JS0 | L8: Test if/else statements 6. JS0 | L10: Code a quiz score 7. Name selector

Algorithms & Programming **Scripting Showdown** 10 hrs

4	JavaScript IDE	Students learn about the arithmetic operators used in JavaScript to perform calculations. They also refresh their knowledge of order of operations (BEDMAS) to ensure the calculation is expressed correctly. They use JavaScript code to create programs that calculate the area of rectangles, triangles, and circles.	1. Unit slides 27 - 36 2. JS0 | L6 Task 4: Using variables to store data 3. JavaScript programs 4. BEDMAS worksheet (answers) 5. Calculate the area of a triangle 6. Calculate area of a circle
5	Flags & Functions	Students are introduced to the concept of a function. These functions can be used to create a range of shapes such as rectangles, triangles, and circles. An offline activity allows them to explore the output of the function. The final activity requires them to design flags using a range of functions and parameters.	1. Unit slides 37 - 47 2. JS103 | Drawing Flags 3. Shape function reference sheet 4. Functions activity (answers) 5. Coloured pencils (optional)
6	Food Frenzy 1	Students create a game that involves positioning items as collectibles or obstacles. They use a language called PaperScript which is a library of JavaScript commands. Students also focus on the syntax of the language, and identify and fix coding bugs.	1. Unit slides 48 - 51 2. JS105 | Food Frenzy 3. Code review worksheet (answers)
7	Food Frenzy 2	Students investigate the use of comparison operators to evaluate conditional statements as true or false. Students complete, customize, and share their games. They also submit their code reflection document as evidence of learning and mid-unit formative assessment.	1. Unit slides 52 - 54 2. Operator cards 3. 2 x packs of playing cards 4. JS105 | Food Frenzy 5. Food Frenzy item list 6. Food Frenzy code extras
8	Python Perfection	Students start by comparing the syntax of basic input and output commands in JavaScript and Python. They then look at Python variable naming rules, create a variable and an input statement, and use the format() function to print (output) the response.	1. Unit slides 55 - 66 2. "What is Python?" video 3. PY0 | L1: Intro & print statements 4. PY0 | L2: Variables and input 5. PY0 Teaching Guide 6. JS0 | Lesson 6 recap 7. Python variables worksheet (answers) 8. Python variable names reference

Algorithms & Programming **Scripting Showdown** 10 hrs

9	Loop Loop Loops	Students create two different types of loops using Python code. A “for” loop that repeats a certain number of times, and a “while” loop that repeats until a condition is false.	1. Unit slides 67 - 71 2. PY0 | L4: Repeating code - ‘for’ & ‘while’ loops 3. PY101 | Python Grid Challenge 4. For Loop extension task 5. "Coding Basics While Loop" video
10	Pack up Python	Students revise programming concept vocabulary by completing a crossword. They complete the PY0 course to ensure they are comfortable with Python code that handles input data using variables, selection, and iteration (for/while loops). The lesson concludes with a Kahoot! to test their knowledge of Python syntax and programming concepts.	1. Unit slides 72 - 73 2. Python crossword (answers) 3. PY0 | L3: If/else statements 4. PY0 | L5: Drawing graphics with Turtles 5. End of unit Kahoot!

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we’d love it if you could fill out as many as you’re able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you’re short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 8 UNIT PLAN

CODE AVENGERS

Algorithms & Programming | **'Rithms 'n' Hues** | 13 hrs

Unit Description

This unit builds on the foundations of JavaScript from the grade 7 unit. Students work through the **PR4: Mind Vs. Machine** online course and create an animated park scene, an LED light sequence, and a dance routine for the school robots.

Their programs include the use of 'for' loops, compound conditionals, and arrays. There is a mini project at the end of the unit that provides the opportunity to create functions with parameters.

Learning Outcomes

Target ages: US/CA Grade 8 | **AU** Year 8 | **NZ/UK** Year 9

- Decompose problems and design efficient, well documented code that is thoroughly tested.
- Plan programs by creating algorithms using flowcharts or pseudocode and use a trace table to test the logic.
- Create RGB colors and consider accessibility features.

Curriculum Links

US:

- **2-AP-10:** Use flowcharts and/or pseudocode to address complex problems as algorithms.
- **2-AP-11:** Create clearly named variables that represent different data types and perform operations on their values.
- **2-AP-12:** Design and iteratively develop programs that combine control structures (nested loops and compound conditionals).
- **2-AP-13:** Decompose problems and subproblems into parts.
- **2-AP-14:** Create procedures with parameters to organize code.
  2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.
- **2-AP-17:** Systematically test and refine programs.
- **2-AP-19:** Document programs.
- **2-IC-21:** Discuss issues of bias and accessibility.

UK:

- **KS3:** Use 2 or more programming languages (one textual); use data structures and develop modular programs that use procedures or functions.
- **KS3:** Understand simple Boolean logic (AND/OR/NOT).

NZ:

- **CT P04:** Create programs with inputs, outputs, sequence, selection. Comparative and logical operators, variables, iteration.
- **DO P03:** Follow a defined process to design, develop, test, and evaluate digital content to address given contexts or issues, taking into account immediate social, ethical, and end-user considerations.

AU:

- **AC9TDI8P05:** Design algorithms involving nested control structures and represent them using flowcharts and pseudocode.
- **AC9TDI8P06:** Trace algorithms to predict output for a given input and to identify errors.
- **AC9TDI8P09:** Implement, modify and debug programs involving control structures and functions in a general purpose programming language.

CA:

- **C-3.1:** Solve problems for mathematical situations including writing code that involves events, a defined count and/or subprogram and other control structures.
- **C-3.2:** Read and alter existing code, and describe how changes to the code affect the outcomes and the efficiency of the code.

GRADE 8 UNIT PLAN

CODE AVENGERS

Algorithms & Programming | **'Rithms 'n' Hues** | 13 hrs

Cross-Curricular Links

English:

- Writing code comments helps students practice technical writing skills and learn how to communicate their ideas clearly and concisely.
- Writing a program requires students to think logically and sequentially about the steps involved in solving a problem. This skill is transferable to writing where students need to think logically about the structure of their writing, such as the order of paragraphs and the flow of ideas.
- Group discussions help students develop their speaking and listening skills.

Math:

- This unit incorporates a number of math concepts such as coordinates (Transform, reflect), probability, negative numbers and measuring angles in degrees.

Art/Design:

- Students learn to communicate their ideas visually through the creation of flowcharts.

#	Lesson Title	Lesson Description	Materials and Resources
1	Sort That Pseudocode	Students revise the concept of reflection of a point on the x and y axis. They discuss the benefits of subprograms (functions/procedures) and sort some pseudocode, which contains 2 subprograms, into order.	1. Unit slides 2. Reflections task (answers) 3. Rotation, Reflection, Translation 4. PR4 | L1: Park scene 5. PR4 Teaching guide 6. Scratch reflect project 7. Pseudocode sort (cut into pieces) 8. Red, black and blue pens
2	Verify Variables	Students learn that a core part of programming is storing information in variables that follow naming conventions. They write code that contains variables to position and move sprites. Students will use conditions to check the x coordinate of the sprite and modify the direction of travel.	1. Unit slides 2. Negative numbers recap (answers) 3. JavaScript IDE 4. Naming variables teacher notes 5. Naming JavaScript variables (answers) 6. PR4 | L2: Animation with onFrame - Tasks 1-3 7. Code review (answers)

Algorithms & Programming | **'Rithms 'n' Hues** | 13 hrs

3	Compound Conditionals	Students identify three logical operators && (AND), || (OR), and ! (NOT) to make more complex conditions and combine expressions to make compound conditionals. They play a game to either snap (play) or skip a music track depending on whether the condition is True or False for matching the person's music taste. They apply their new skills by creating an 'if' statement containing the OR operator.	1. Unit slides 17-29 2. "CS Principles: Conditionals - Part 3" video 3. "Shuffle, Play a Random YouTube song" 4. Spotify 5. Set of Snap Skip cards 6. Whiteboards or paper 7. PR4 | L2: Animation with onFrame - Tasks 4-5 8. JavaScript IDE 9. Exit ticket (answers)
4	Recommender Systems	Students consider online services that recommend things to them to watch, buy, or listen to. They explore recommender systems in more depth and look at two different types of filtering. Students spend the second part of the lesson searching for a dataset for a recommender system they will design later in the unit.	1. Unit slides 31-39 2. Padlet/Answer Garden 3. "Introduction to Recommender Systems" video 4. Dataset collection sheet 5. IMDb Top 250 Movies 6. Kindle best sellers 7. Nike
5	Randomness	Students start by considering the probability of a certain piece of data being selected through the use of spinning wheels and dice which produce names and numbers at random. The concept of random numbers is explored further through the creation of a JavaScript function. The skills learned here are applied to the forest animation, where raindrops appear in random positions.	1. Unit slides 40 - 46 2. JavaScript IDE 3. PR4 | L3: Random raindrops 4. Probability Wheel 5. Code review questions
6	Pair Programming	Students learn about the benefits of pair programming and put these skills into practice in a code bug relay task. The elements of functions are explored (parameters/arguments) and students can experiment with the values of a rectangle function. They also change opacity and hue values when creating a lighting scene.	1. Unit slides 47-54 2. "Pair Programming" video 3. Relay Tasks Sheet 4. Relay Answer Sheet 5. Highlighter pen/pencil 6. JavaScript IDE 7. PR4 | L4: Shapes & keyboard events 8. Exit ticket (answers)

Algorithms & Programming **'Rithms 'n' Hues** 13 hrs

7	Know How To Nest	Students analyze flowcharts with multiple decisions and compare this to a computer program with nested control structures. They explore the use of an 'else if' statement. This allows an input to be checked against multiple conditions. The lesson concludes with a Kahoot to review the first half of the PR4 course.	1. Unit slides 55 - 61 2. Are you ready for a dog flowchart 3. Should I get a dog flowchart 4. Should I get a dog code 5. Flowchart match recap activity 6. JavaScript IDE 7. PR4 | L5: Scene changes: else if 8. Kahoot! Mid-course review
8	For Loops	Students study a flowchart which shows 5 concurrent light sequences for the LED lights attached to the robot costumes. They discuss different aspects of the flowchart in groups. Students then use this flowchart to predict what the lighting sequence output will be. Finally they program the lighting sequence by using 'for' loops to repeat parts of the sequence.	1. Unit slides 62 - 66 2. LED flowchart 3. LED flowchart questions (answers) 4. Robot LED predictions (answers) 5. PR4 | L6: Loops and LED wearables 6. Exit ticket (answer on slide 71) 7. Colored pencils
9	Robot Routines	Students recap angles and turns so they are familiar with 90, 180, and 270 degree turns. They create repeated sequences with turns to create a robot dance routine. Students identify a range of different bug types and reproduce similar ones for their partner to find. The teacher also demonstrates debugging using the live coding technique.	1. Unit slides 67 - 70 2. PR4 | L7: Dancing robots 1 3. PR4 | L8: Dancing robots 2
10	Trace Tables	Students are taught how to use trace tables to record input, variables, output values, and other information to test and analyze an algorithm, so they can observe how the computer executes it step by step. The second half of the lesson focuses on students developing an algorithm to recommend a product.	1. Unit slides 74 - 78 2. Using trace tables 3. Fizz Buzz 4. Shall I get a dog pseudocode & trace table 5. Dog breed cards 6. Diagrams.net for flowchart creation. 7. Paper, ruler, eraser

Algorithms & Programming **'Rithms 'n' Hues** 13 hrs

11	Color Vision	Students start by recapping how the different components of the eyes work together to see color. They watch a video about color blindness and explore a range of resources related to this topic. The second half of the lesson involves writing code to generate different colors and concludes with a fun game of Bingo!	1. Unit slides 79 - 80 2. How the eye works 3. PR4 | L9: Stage lighting 4. "Color Vision Deficiency" video 5. Explore color vision deficiencies 6. Bingo cards 7. RGB calling cards 8. Pen/highlighter/marker/counters
12	Hurray Array	Students learn that arrays are useful for holding large numbers of related values. They learn the syntax of arrays and create arrays to hold sound effects and music filenames. At the end of the lesson, students wrap up with a fun game of PictionArray which involves drawing images and, when the correct word is guessed, looking it up in an array sheet and supplying the correct index.	1. Unit slides 81 - 86 2. Makecode example 3. PR4 | L10: Sound and music 4. PictionArray arrays (students) 5. PictionArray values (teacher) (printed back to back and cut into pieces) 6. IDE
13	Parameters Please	Students start the lesson which shows them how they can reuse code (with different inputs) to produce multiple outputs. The benefits of functions are explored through the analogy of a milk processing plant. Students learn the syntax of a function that uses parameters. They also revisit inbuilt functions such as Path.Star.	1. Unit slides 87 - 91 2. Pass the parameter cards 3. Silly stories program | Code 4. Times Table code 5. Path.Rectangle | Path.Circle | Path.Star

Unit feedback/self-assessment survey links on next page

Algorithms & Programming | **'Rithms 'n' Hues** | 13 hrs

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

GRADE 8 UNIT PLAN

CODE AVENGERS

Programming Project

Party Parameters

8 hrs

Unit Description

This 8 lesson project designing digital confetti allows students to apply their JavaScripts skills learned in previous units. They recap JavaScript function parameters and explore the feasibility of design ideas within the given time frame. Students follow an iterative process and emphasis is placed on systematic testing and identifying opportunities for refinement. Overall, this unit provides a comprehensive hands-on experience in JavaScript programming and project development skills.

Learning Outcomes

Target ages: US/CA Grade 8 | **AU** Year 8 | **NZ/UK** Year 9.

- Develop a program using an iterative process involving design, implementation, and review.
- Create a function with parameters to control the quantity, size, color, and position of generated shapes.
- Systematically test and refine a program.

Curriculum Links

US:

- **2-AP-10:** Use flowcharts and/or pseudocode to address complex problems as algorithms.
- **2-AP-11:** Create clearly named variables that represent different data types and perform operations on their values.
- **2-AP-12:** Design and iteratively develop programs that combine control structures (nested loops and compound conditionals).
- **2-AP-13:** Decompose problems and subproblems into parts.
- **2-AP-15:** Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
- **2-AP-14:** Create procedures with parameters to organize code.
- **2-AP-17:** Systematically test and refine programs.
- **2-AP-19:** Document programs.

AU:

- **AC9TDI8P05:** Design algorithms involving nested control structures and represent them using flowcharts and pseudocode.
- **AC9TDI8P06:** Trace algorithms to predict output for a given input and to identify errors.
- **AC9TDI8P09:** Implement, modify and debug programs involving control structures and functions in a general purpose programming language.

NZ:

- **CT P04:** Create programs with inputs, outputs, sequence, selection. Comparative and logical operators, variables, iteration.
- **DO P03:** Follow a defined process to design, develop, test, and evaluate digital content to address given contexts or issues, taking into account immediate social, ethical, and end-user considerations.

UK:

- **KS3:** Use 2 or more programming languages (one textual); use data structures and develop modular programs that use procedures or functions.
- **KS3:** Understand simple Boolean logic (AND/OR/NOT).

CA:

- **C-3.1:** Solve problems for mathematical situations including writing code that involves events, a defined count and/or subprogram and other control structures.
- **C-3.2:** Read and alter existing code, and describe how changes to the code affect the outcomes and the efficiency of the code.

Programming Project

Party Parameters

8 hrs

Cross-Curricular Links

English:

- Write code comments to practice technical writing skills and learn how to communicate ideas clearly and concisely.
- Write a program by thinking logically and sequentially about the steps involved, and relate this to the need to think logically about the structure of writing, such as the order or paragraphs and flow of ideas.
- Develop speaking and listening skills through group discussions.

Math:

- This unit incorporates a number of math concepts such as coordinates (transform, reflect), probability, negative numbers and measuring angles in degrees.

Art/Design:

- Communicate ideas visually through the creation of flowcharts.

Social Science:

- Students learn that pollution and waste can have a negative impact on the environment, and they can learn about ways to reduce or eliminate these.

#	Lesson Title	Lesson Description	Materials and Resources
1	Project Launchpad	This lesson introduces students to an 8 lesson project where they will be designing their own digital confetti using an iterative design process. They recap how to write a function with parameters in JavaScript.	1. Unit slides 2-13 2. Project workbook (p1-4) 3. JavaScript IDE 4. Exit ticket 5. JS103	Flags Project tasks 1-5
2	Prototyping Playground	This lesson is an opportunity for students to try out design ideas to see what is feasible in the time available. Basic functionality such as randomization is revisited so shapes can be different sizes and positions.	1. Unit slides 14-18 2. Project workbook p5 3. "What is a prototype?" video 4. JavaScript IDE 5. Paper.js - Path reference 6. Flags project tasks 9-10
3	Breaking It Down To Build It Up	Students are prompted to consider which values can be obtained from the user and how they can be gathered, as well as how these values can be utilized within the program.	1. Unit slides 19-23 2. Project workbook p6 3. Coolors.co 4. JavaScript IDE 5. Padlet

Programming Project

Party Parameters

8 hrs

4	Fabulous Flowcharts	Students explore the flowchart shapes used to visually represent key programming structures. All students will attempt to create a flowchart to show the steps needed to create a singular confetti shape. Some students may plan out how they will gather user inputs to pass values to the confetti shape function.	1. Unit slides 24-26 2. Project workbook p7 3. Flowchart Kahoot! 4. Diagrams.net flowchart creator 5. Pencil, paper, ruler 6. How to Make a Flowchart on diagrams.n…
5	Tenacious Testing	Students learn about the difference between syntax and logic bugs. They have the opportunity to locate syntax bugs and use a test plan to identify logic bugs. Students are encouraged to use systematic testing, whilst developing their confetti solution, to gain an *'exceeding'* grade on the project.	1. Unit slides 27-29 2. Project workbook p9-10 3. Spot the syntax bugs (answers) 4. Animal antics code 5. Testing plan (answers)
6	Usability Testing	Students are introduced to the concept of usability testing. They are encouraged to make a list of tasks they would like the user to perform and questions they would like to ask users. Students then take any feedback from users as an opportunity to refine their confetti solution.	1. Unit slides 30-32 2. Project workbook p11 3. Usability Testing in UX Design Thinking P… 4. Usability testing 101 5. Testing plan ideas
7	Review, Improve	Students recap some key vocabulary from the unit through the completion of a wordsearch. They are encouraged to conduct some usability testing and refine their solution based on the feedback they receive.	1. Unit slides 33-34 2. Word quest (answers) 3. Project workbook p11 (Step 7) 4. Wheel of names
8	Excel At Evaluations	In this last lesson, students prepare to evaluate their project performance. Firstly they explore two methods used in stand-up meetings: Kanban boards and "3 questions". These methods help them reflect on any obstacles they have faced and identify ways to overcome them.	1. Unit slides 35-37 2. Project workbook p11 3. Digital Profession Plays – Stand-ups 4. Post-its 5. Kanban board handout 6. Self assessment survey (printable) 7. Student feedback to CA survey

Unit feedback/self-assessment survey links on next page

Programming Project **Party Parameters** 8 hrs

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Student to CA Unit Feedback:** This is where you ask your students to tell us how they found the unit. You may need to provide some guidance on when the learning for this unit started and ended (what topics/activities).
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.

KINDERGARTEN UNIT PLAN

CODE AVENGERS

Data Representation | **Data Detectives** | 10 hrs

Unit Description

Students work through the DR1: Carnival Crisis online course. They attempt various tasks to learn to categorize and order data. They are introduced to a range of measurement units including height, weight and temperature. They explore a deck of cards and play a card match game. All of the online skills are reinforced in an offline context such as physically sorting and weighing objects. Students get numerous activities to collect data such as a treasure hunt style activity and surveying each other.

Learning Outcomes

Target ages: US/CA Kindergarten | **AU** Foundation/Prep | **NZ/UK** Year 1.

- Sort objects and shapes by a single feature.
- Develop comparative language skills.
- Collect and visualize data in two different forms (pictographs and tally charts).
- Predict missing sections in patterns.

Curriculum Links

US:

- **1A-DA-05:** Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data.
- **1A-DA-06:** Collect and present the same data in various visual formats.

AU:

- **ACTDIFK02:** Represent data as objects, pictures and symbols.
- **ACMMFST01:** Collect, sort and compare data represented by objects and images in response to given investigative questions that relate to familiar situations.

NZ:

- **CT D01:** They know how to use some applications, they can identify the inputs and outputs of a system, and they understand that digital devices store content, which can be retrieved later.

UK:

- **KS1 CO4:** Use technology purposefully to create, organize, store, manipulate and retrieve digital content.

Cross-Curricular Links

Math:

- Number patterns (5, 10).
- Mass - compare weights, use comparative language.
- Length - compare the length of two objects.
- Temperature - describe objects as hot/cold, order 2+ objects.
- Patterns - predict missing sections.
- Statistics - gather, display and/or count category data.

English:

- Speaking - adapt communication for different purposes/audiences.
- Reading - understand and interpret a range of texts.

Science:

- Living World - human body and senses.
- Physical World - heat.

KINDERGARTEN UNIT PLAN

CODE AVENGERS

Data Representation | **Data Detectives** | 10 hrs

#	Lesson Title	Lesson Description	Materials and Resources
1	Super Senses	This lesson introduces the senses of sight, taste, touch, and smell, discussing how they relate to carnival or fair experiences. Students engage with interactive activities, share their carnival experiences, and play a game where they describe senses to help others guess a location.	1. Unit slides 1-4 2. DR1 | L1 Collect data 3. Blindfolds (optional) 4. Food e.g. cotton candy (optional) 5. "Carnival sound effects" video 6. "The Five Senses" video 7. Where am I? cards
2	Dig That Data	Students consider what the word 'data' means and how it can be collected using our senses, questioning, or measuring. Students engage in a data hunt mission to find different types of data. In the second half of the lesson they are introduced to the features of the Code Avengers JR platform and they work through the first 3 tasks of lesson 2 themselves. They gather as a class to review the content in the last 2 tasks.	1. Unit slides 5-10 2. DR1 | L2 Sorting data 3. Data mission cards 4. "Quick Login" video 5. Pair programming information sheet
3	The Temperature Tour	Students start the lesson by looking at a range of foods from across the world. They sort these into two groups: hot and cold foods. The concept of temperature is explored further when students look at weather temperatures in different cities. They sort the data into order.	1. Unit slides 11-14 2. DR1 | L3 Organizing the food stalls 3. Food cards & scissors 4. City temperature cards 5. Day and night world map 6. Thermometer 7. World Map/Globe
4	Height Heroes	Students start the lesson by sorting small groups of students into height order. They orally describe the height of students using comparative language such as shorter and tallest. They practice using this language in written form. They measure height and width of the clowns during completion of the Carnival Crisis online lesson 4.	1. Unit slides 15-19 2. DR1 | L4 Sorting the clowns and stalls 3. Glue/scissors/pencils 4. Savannah sizing 5. Clowning around worksheet (answers) 6. Measuring tapes 7. "What's your APE index?" video

Data Representation **Data Detectives** 10 hrs

5	Petting Pictographs	This lesson builds students skills in interpreting and comparing data visually. They are introduced to the idea of a pictograph and create their own one about their favorite carnival rides. They practice using pictographs on the online Carnival Crisis course so they can visualize the number of animals that need to be moved to the pens.	1. Unit slides 20-22 2. DR1 | L5 The petting zoo 3. Post-it notes 4. Coloring pencils 5. Colorful dot stickers 6. Pictograph worksheet 7. Animal themed books
6	Tally Ho!	In this lesson dot plots are introduced as an alternative to pictographs which were covered last lesson. Dot plots can be used to count and compare things. Tally marks are explained as a quicker way to collect and record data, and students practice using them. They gather tally marks from others using speed surveys. Finally, they count their tally marks and discuss popular responses.	1. Unit slides 23-25 2. DR1 | L6 Dot plots and graphs 3. Tally task worksheet 4. Speed survey
7	Wondering About Weight	Students learn about weight and how to compare whether objects are heavier or lighter than one another. They participate in interactive activities, such as using a seesaw or balance scale, feeling and comparing the weights of different objects, and estimating the weight of everyday objects	1. Unit slides 26-29 2. DR1 | L7 Comparing weights 3. "Is it heavy or light" 4. Compare weights worksheet 5. Objects (can, pasta, apple, teddy, etc.) 6. Balance scales
8	Suits You!	Students discuss card games they've played. They explore the concept of suits and symbols on cards. Students form groups based on their card suits and arrange themselves in order. They complete online tasks to practice creating card patterns in lesson 8 of the Carnival Crisis course.	1. Unit slides 30-32 2. DR1 | L8 The magician's pattern 3. Packs of cards 4. Suit group labels 5. Random card selector
9	Repeating Patterns	Students start the lesson by discussing different types of patterns e.g. numbers, sounds, and colors. They complete a number of repeating pattern challenges whilst working through lesson 9 of the Carnival Crisis course. There is also the opportunity to play a memory match card game.	1. Unit slides 32-33 2. DR1 | L9 Testing the arcade games 3. we will rock you instrumental video 4. Shape patterns worksheet 5. Coloring pencils 6. Carnival match cards 7. Playing cards

Data Representation **Data Detectives** 10 hrs

10	Making Money	Students start by practicing subtracting 1, 2, 3, and 4 from numbers 10 or less. They are introduced to the idea of place value through carnival money whilst completing the last lesson of the online course, Carnival Crisis. They design their own money, create a shop, and apply their subtraction skills by giving change.	1. Unit slides 36-37 2. DR1 | L10 Counting money 3. Subtraction skills worksheet 4. One hundred worksheet 5. Play money sheets & Bank note symbols 6. Counting chips 7. Coloring pencils, scissors, glue 8. Items for play shops, paper for price labels 9. Surveys

Unit Surveys

The links below are to the surveys for this unit. We know teachers are tight on time, but we'd love it if you could fill out as many as you're able to in order to help us refine these resources and make them the best they can be! The surveys are as follows:

- **Teacher to CA Unit Feedback:** This is where you tell us how this unit went overall.
- **Teacher to CA Lesson Feedback:** This form can be filled out for each lesson in this unit, but if you're short on time, then feedback on any problematic, or particularly successful lessons/activities would also be great.
- **Student to Teacher Self-Assessment:** This form is for students to tell you how they are doing with the learning outcomes for the unit. You can make a copy and give to the learners at the end of the unit, or before and after for PRE/POST formative/summative assessment. Here is a printable Google Doc version if you prefer.