Progress Outcome	Progress Outcome	Kids Speak
1	In authentic contexts and taking account of end-users, students use their decomposition skills to break down simple non-computerised tasks into precise, unambiguous, step-by-step instructions (algorithmic thinking). They give these instructions, identify any errors in them as they are followed, and correct them (simple debugging)	• I look at a task and can tell others the order of steps they need to follow (create instructions). • I can give the instructions to others. • I can see the mistakes in my instructions when others follow them and I can fix them.
2	In authentic contexts and taking account of end-users, students give, follow, and debug simple algorithms in computerised and non-computerised contexts. They use these algorithms to create simple programs involving outputs and sequencing (putting instructions one after the other) in age-appropriate programming environments.	• I can give instructions. • I can follow the instructions from other students. • I can find mistakes in the instructions and fix them (debugging). • I can use my instructions to create a computer program.

Developed by Monika Kern and Tara Fagan, Raranga Matihiko

Attribution-NonCommercial-ShareAlike
CC BY-NC-SA

3	In authentic contexts and taking account of end-users, students decompose problems into step-by-step instructions to create algorithms for computer programs. They use logical thinking to predict the behaviour of the programs, and they understand that there can be more than one algorithm for the same problem. They develop and debug simple programs that use inputs, outputs, sequence, and iteration (repeating part of the algorithm with a loop). They understand that digital devices store data using just two states represented by binary digits (bits).	• I can look at a problem and can figure out the steps to solve it. I use this solution to code algorithms for computer programs. • I can predict what my computer program will do based on the code it receives. • I understand there can be different algorithms that solve the same problem. • In my code I will use *inputs, outputs, sequence* and repeat loops (*iteration*). • I debug my computer program so it runs efficiently. • I know that computers use bits to store data.

Developed by Monika Kern and Tara Fagan, Raranga Matihiko

Attribution-NonCommercial-ShareAlike
CC BY-NC-SA

4	In authentic contexts and taking account of end-users, students decompose problems to create simple algorithms using the three building blocks of programming: sequence, selection, and iteration. They implement these algorithms by creating programs that use inputs, outputs, sequence, basic selection using comparative operators, and iteration. They debug simple algorithms and programs by identifying when things go wrong with their instructions and correcting them, and they are able to explain why things went wrong and how they fixed them. Students understand that digital devices represent data with binary digits and have ways of detecting errors in data storage and transmission. They evaluate the efficiency of algorithms, recognising that computers need to search and sort large amounts of data. They also evaluate user interfaces in relation to their efficiency and usability	• I can solve a problem by coding simple algorithms. My algorithms use *sequence*, *selection* and *iteration*. • I use the algorithms to make computer programs that use *inputs*, *outputs, sequence, iteration,* and *comparative operator*s. • To debug algorithms and computer programs I find the mistakes in my algorithms and correct them. I can explain why things went wrong and how I fixed them. • I know that digital devices use 0s and 1s (*binary*) to represent information. • I know that a digital device will check files and recognise when they are *corrupted*. • I can look at an algorithm and check if it is efficient to help a digital device search and sort information better. • I can look at user interfaces and check if they are user friendly and efficient.

Developed by Monika Kern and Tara Fagan, Raranga Matihiko

Attribution-NonCommercial-ShareAlike
CC BY-NC-SA

5	In authentic contexts and taking account of end-users, students independently decompose problems into algorithms. They use these algorithms to create programs with inputs, outputs, sequence, selection using comparative and logical operators and variables of different data types, and iteration. They determine when to use different types of control structures. Students document their programs, using an organised approach for testing and debugging. They understand how computers store more complex types of data using binary digits, and they develop programs considering human-computer interaction (HCI) heuristics	• I can solve a problem by coding simple algorithms without teacher help. • I use the algorithms to make computer programs that use *inputs, outputs, sequence, iteration, comparative and logical operators* and *variables of different data types*. • I decide what kind of *control structures* to use in my code and when to use it. • I add comments to my code to explain how it works or what it does. • I keep a record of all the work on my programs including testing and debugging. • I know that computers store information using *binary*. • When I code my programs, I think of how the end-user will use them, and I develop, test and refine my programs to meet their needs. • When I design and evaluate user interfaces, I follow good practices *(heuristics)* so my program can be both effective and user friendly.

Developed by Monika Kern and Tara Fagan, Raranga Matihiko

Attribution-NonCommercial-ShareAlike
CC BY-NC-SA

References:

Ministry of Education. (2017). Technology in the New Zealand Curriculum. Wellington, New Zealand: Author.

Additional Links:

Kidspeaking the literacy progressions

Digital Technology Glossary

Additional Support:

Technology online - http://technology.tki.org.nz/

Kia Takatū ā-Matihiko | Digital Readiness - https://kiatakatu.ac.nz/

DTHM for Kaiako - https://www.dthm4kaiako.ac.nz/

Raranga Matihiko | Weaving Digital Futures empowers ākonga/kaiako to think, challenge and create using authentic integrated experiences. We work with NZ schools and Kura to provide opportunities for tamariki through digital technologies. We are based in four locations – Waitangi, Auckland, Hawkes Bay and Wellington and are run through our partnership museums Te Papa, Auckland Museum, MTG Hawkes Bay and Te Kōngahu Museum of Waitangi.

We currently cater to schools that are decile 1-3 and are within a 100km of the museums and all kura within a 120km radius as part of the Ministry of Education Digital Technology for All Equity Fund.

The resources and information on our website are available to all teachers, kaiako and whānau.

Developed by Monika Kern and Tara Fagan, Raranga Matihiko

Attribution-NonCommercial-ShareAlike
CC BY-NC-SA

Developed by Monika Kern and
Tara Fagan, Raranga Matihiko

Attribution-NonCommercial-ShareAlike
CC BY-NC-SA