

**Internal Assessment Resource**

**Digital Technologies & Hangarau Matihiko Level 1**

This resource supports assessment against Achievement Standards 91883 and 91884[[1]](#footnote-1)

**Standard title:** Develop a computer program (4 credits)

Use basic iterative processes to develop a digital outcome (6 credits)

**Credits:** 10

**Resource title:** Chatbot Charlie

**Resource reference:** Digital Technologies & Hangarau Matihiko 1.7B\_1.8B Version 2

|  |
| --- |
| This resource:* Clarifies the requirements of the achievement standard
* Supports good assessment practice
* Should be subjected to the school’s usual assessment quality assurance process
* Should be modified to make the context relevant to students in their school/kura environment and ensure that submitted evidence is authentic
 |

|  |  |
| --- | --- |
| Date version published by Ministry of Education | March 2019 Version 2To support internal assessment from 2018 |
| Authenticity of evidence | Teachers/kaiako must manage authenticity for any assessment from a public source, because students/ākonga may have access to the assessment schedule or student exemplar material.Using this assessment resource without modification may mean that students’ work is not authentic. The teacher may need to change figures, measurements or data sources or set a different context or topic to be investigated or a different text to read or perform. |

**Internal Assessment Resource**

**Achievement standard:** 91883 and 91884

**Standard title:**  Develop a computer program

Use basic iterative processes to develop a digital outcome

**Credits:** 10

**Resource title:** Chatbot Charlie

**Resource reference:** Digital Technologies & Hangarau Matihiko 1.7B\_1.8B Version 2

**Teacher/Kaiako guidelines**

The following guidelines are supplied to enable teachers/kaiako to carry out valid and consistent assessment using this internal assessment resource.

Teachers/kaiako need to be very familiar with the outcome being assessed by the achievement standards. The achievement criteria and the explanatory notes contain information, definitions, and requirements that are crucial when interpreting the standard and assessing students/ākonga against it.

**Context/Te Horopaki**

This activity requires students to use basic iterative processes to develop a refined chatbot program. The chatbot must be able to hold a simple conversation by recognising words in the input sentences and providing a suitable reply.

Students must manage the development of their program by decomposing it into components. The components must be trialled in an iterative manner. They must trial multiple components and/or techniques and select the most suitable. They then must apply this information to develop a high-quality outcome. Students will also need to describe and address relevant implications. The program must be documented with variable names comments and which describe how the code works. Students must follow conventions for the language that they are using. Their program must be comprehensively tested and debugged. *It is important that students show evidence of testing against boundary cases.* The program must be flexible and robust, and should be a well-structured, logical response to the task.

The chatbot will chat by prompting the user and responding to the user’s input. The chatbot will detect keywords in the input and use these to guide its reply. Any input outside of expected input might cause the chatbot to present a new question. Simple, focussed topics, with a limited range of responses, work best to begin with and enjoy the most success (for example a subject like sport, or Māori phrases for English words, or Rogerian psychotherapy), however, the final program could develop more complex input analysis, basic memory, sentence reconstruction etc. This provides an almost unlimited scope to extend more able students. See the resources section for further information.

At Level 1, the chatbot can be quite limited in its responses, and may rely mainly on pre-defined output. A general chatbot that responds to a wide range of inputs, analyses the sentence structure of the input or implements learning algorithms is beyond Level 1 and therefore not expected for this assessment.

Students should be encouraged to develop reusable code. For example, the chatbot’s response sentences might be held in a list, and an appropriate one selected. By replacing the sentences, the chatbot’s interactions can be easily changed. Students should think about the data structures and the functions they will use to achieve this.

Students should identify the components of their chatbot, and how these might be iteratively developed to improve and extend the behaviour of the program. Students should identify different components and techniques which they can trial, in order to select the most suitable.

Students should follow conventions for the language that they are using.

To prepare students for this assessment teachers should encourage students to examine conversations, conversation starters and the difference between open and closed questions. This will assist students to create the correct sentence sets to allow their chatbot to hold a reasonable conversation on a fixed topic.

***Evidence***

Assess students on:

* how they have described relevant implications, and the extent to which they have addressed them
* how comprehensively the program has been tested and debugged; whether they have tested expected, boundary and invalid cases
* how they have followed conventions for the language that they are using
* how they have documented the program with variable names and comments
* how flexible and robust the program is, and whether it is well-structured
* how they have managed the development of their program, by decomposing it into components
* how they have trialled the components of their program
* their application of an iterative development process to improve and refine the components of their computer program
* the extent to which their computer program has applied information from their planning, testing, and trialling of components to ensure the outcome is of a high quality

Students must develop their own program, including the questions and responses that their chatbot produces. They must ensure that they have described and addressed relevant implications, for example, using appropriate cultural and ethical chatbot responses.

Students could keep a succinct development log where they record regular reflections on their progress. This could include written evidence, annotated screenshots, diagrams, short video clips, or code snippets. You may want to give students guidance on appropriate style and format for their development log. Quality is more important than quantity. These achievement standards do not assess format or style of the evidence.

They should be encouraged to regularly save versions of code as they work towards the final outcome. Teachers could use this opportunity to teach and use version control software such as Git or CVS to help gather assessment evidence although this is not a necessity.

Students will need good working knowledge of the chosen programming language before attempting this project and time should be allocated at the start of the teaching and learning to ensure this is the case. They should be able to use:

* variables and constants
* basic input and output commands
* sequence, selection and iteration control structures
* lists, arrays, dictionaries and/or functions, methods, procedures
* other built-in functions

**Conditions/Ngā Tikanga**

Students may share ideas and concepts by working collaboratively but must implement the code on their own. All assessed code should be written by individual students and be their own work. Any code that is not their own must be acknowledged as such and will not be used to determine their grade.

Conditions of Assessment related to this achievement standard can be found at <http://ncea.tki.org.nz/Resources-for-Internally-Assessed-Achievement-Standards>

**Resource requirements/Ngā Rauemi**

Students will need access to hardware and software necessary to produce, test and trial the outcome, and to record any development log.

*Links to assist:*

<https://wiki.scratch.mit.edu/wiki/Artificial_Intelligence>

<https://www.youtube.com/watch?v=gZAdYyjxRcg>

<https://medium.freecodecamp.org/creating-a-chat-bot-42861e6a2acd>

<https://www.chatbots.org/ai_zone/viewthread/492/>

<https://www.codeproject.com/Articles/36106/Chatbot-Tutorial>

<https://chatbotsmagazine.com/design-framework-for-chatbots-aa27060c4ea3>

<https://apps.worldwritable.com/tutorials/chatbot/>

<https://venturebeat.com/2016/07/27/5-tips-for-writing-chatbot-scripts/>

<http://hawkee.com/snippet/6820/> (Magic 8-ball program)

*Conversation resources:*

<http://www.wikihow.com/Have-a-Great-Conversation>

<http://www.improveyoursocialskills.com/conversation>

<https://www.wikihow.com/Ask-Open-Ended-Questions>

<http://changingminds.org/techniques/questioning/open_closed_questions.htm>

**Additional information/He Kōrero Atu**

***Teacher support information***

The teaching and learning of the knowledge, concepts and skills required to prepare students for this assessment, plus the assessment itself, would occupy a major part of the year’s programme. The total of 10 credits for assessment indicates that approximately 100 hours of teaching, learning (in and out of class) and assessment in a programme of study will be required – about half the year’s programme. Schedule regular progress checks with the students during this activity.

***Planning, trialling and testing:***

Students should present a plan for their outcome.

Students should develop test cases to test the code for every component. Students should provide evidence that they have tested the code and any functions that they have written. Evidence of this testing can be recorded in their development logs. This evidence should include how they have tested against boundary cases.

Students should present evidence of how they have trialled components.

Students should describe a range of relevant implications and show how the final outcome addresses these.

Components of the outcome should be trialled in an iterative manner. For each iteration for the component, students should outline what they have done so far, what they have found difficult and what the next steps should be to produce a more refined outcome

***Final outcome:***

Students use basic iterative processes to develop a refined chatbot program. Teachers should ensure the rigour of the outcome is appropriate for Level 6 of the National Curriculum and that the outcome has not been produced by modifying pre-existing code, or copying a tutorial.

|  |
| --- |
| **Important note to teachers:****Students must test against boundary cases and show evidence of this testing in their submitted documentation.** |

**Internal Assessment Resource**

**Achievement standard:** 91883 and 91884

**Standard title:**  Develop a computer program

 Use basic iterative processes to develop a digital outcome

**Credits:** 10

**Resource title:** Chatbot Charlie

**Resource reference:** Digital Technologies & Hangarau Matihiko 1.7B\_1.8B Version 2

**Student/Ākonga instructions**

**Introduction/Kupu Arataki**

This assessment activity requires you to use basic iterative processes to develop a refined chatbot program. The chatbot must be able to hold a simple conversation by recognising words in the input sentences and providing a suitable reply.

You are going to be assessed on how well you develop your refined program. In developing your program, you must show evidence of basic iterative processes. You will break the outcome into components and trial these components iteratively.

You must **describe** and **address** any relevant implications.

You must show how you have applied information from planning, testing and trialling of components to improve the quality of your program. Your final computer program must be well-structured, robust and logical.

Teacher note: Insert due dates and timeframes

*You may share ideas and concepts, but you must implement the code on your own. Any code that is used from another source must be acknowledged and cannot be assessed as your own work.*

**Task/Hei Mahi**

A chatbot is a computer program that is designed to have a conversation with human users.

You are going to use basic iterative processes to develop a refined chatbot program. You will decompose the program into components and iteratively trial them to improve the quality of the program.

To start this task, you need to:

* Research chatbots and ways to hold a conversation appropriately using open and closed questions
* Think about and describe how your chatbot will address a problem, meet a need, opportunity or interest
* Think about what data structures and/or functions you will require for your chatbot program.
* Save versions of your code as you develop your program. Start up and keep a development log. The development log could include:
	+ Planning - A basic plan of showing the decomposition of the program into smaller components and what the code should achieve and how you might go about achieving it. For example, pseudocode, flowcharts, diagrams or a simple written description. Ensure you have thought about how to test the code (specific test cases).
	+ Development - trial the individual components of the outcome iteratively. Include your code, code comments and adhere to coding conventions. This can be the actual component file saved as a version or a screenshot of the code for the stage.
	+ Trialling and testing - You could record this information in a written paragraph, test table, screen cast or annotated screenshots. For testing, include what you expect to happen and the actual outcome. You should record the results of trialling multiple components and/or techniques. You should outline what the next steps will be and why you have selected your chosen options. You should see an improvement of the components of your program as you develop further iterations.
	+ Evidence that your program functions as intended.

*Planning*

* Plan your outcome – this will involve describing the algorithmic structure of the program, and how the user interacts with it. Your planning must show how you have decomposed your outcome into smaller components
* Plan how you will test your program – this will involve stating what the test cases for expected values are, what are the boundary cases, and how you will comprehensively test your code

*Relevant implications*

* Describe a range of implications that are relevant to your outcome. Include evidence of how you have addressed these in the process of developing the outcome. For example:
	+ Describe the copyright or intellectual property issues that relate to your outcome? What does this mean in relation to your outcome? How did you address these in your completed outcome?
	+ Describe the cultural and ethical implications that relate to your outcome? What does this mean in relation to your outcome? How did you address these in your completed outcome?
	+ Describe why you need to make sure your outcome is usable and functional? What does this mean in relation to your outcome? How did you address these in your completed outcome?

*Development*

* Develop your program through the application of appropriate programming tools and techniques. Your program must demonstrate use of:
	+ variables storing at least two types of data (e.g. numeric, text, Boolean)
	+ sequence, selection and iteration control structures
	+ input from a user, sensors or another external source

**and one or more of:**

* + data stored in collections (e.g. lists, arrays, dictionaries)
	+ user-defined methods, functions or procedures.

Your program must use two or more functions, or two or more lists. It may use both functions and lists.

You should:

* Ensure that you follow conventions for the language, and that you comment your code to describe how it works
* Ensure that your code is flexible and robust
* Iteratively trial the components
* Trial multiple components and/or tools and techniques and refine your outcome based on evidence of your trialling, and select the ones that will work best for the purpose of the outcome
* Carry out testing to improve and refine your outcome. Show evidence that you have applied appropriate testing and debugging procedures to ensure that your program functions as intended
* Show evidence of testing against boundary cases
* Apply the information you have gained from the planning, testing and trialling of components to develop a refined, high-quality outcome.

**Hand in:**

* Your development log including evidence of planning, developing, trialling, testing of your outcome and its components. You must show how you have applied the information from the planning, testing and trialling of components to develop your high-quality outcome. Ensure you have included how you have described and addressed relevant implications.
* The final computer program, and the versions created as you developed it.

Check with your teacher how much information is required in your development log. Quality is more important than quantity.

**Resources/Ngā Rauemi**

**Example of Chatbot conversation where Charlie talks about sport:**

Charlie: Hi

Me: Hi

Charlie: Do you like sport? Yes or No

Me: Yes

Charlie: Do you prefer rugby or tennis?

Me: rugby

Charlie: Do you support the Crusaders or the Hurricanes?

Me: I don’t like either

Charlie: Have you watched the All Blacks play recently?

**Note:** Here Charlie offers the user input choices. Charlie will only respond to these choices but has a few different possible responses to each of the input options. Charlie will also start a new conversation if the expected answer is not received. In this way, there is some variation in the conversations.

**Assessment schedule/Mahere Aromatawai: Digital Technologies & Hangarau Matihiko** **91883 – Chatbot Charlie**

|  |  |  |
| --- | --- | --- |
| **Evidence/Judgements for Achievement/Paetae** | **Evidence/Judgements for Achievement with Merit/Kaiaka** | **Evidence/Judgements for Achievement with Excellence/Kairangi** |
| Develop a computer program. The student has:* written code for a program that performs a specified task using a suitable programming language

**For example (partial evidence)**The student has written a simple, functioning chatbot program in the language of their choice which includes variables of at least two data types, sequence, selection and iteration control structures and input from a user, sensors or another external source. *The chatbot works on a sample of expected cases. The program has either more than one function, or more than one list (or possibly both).*The student has:* set out the program code clearly and documented the program with comments

**For example (partial evidence)***Comments are present but may not be particularly descriptive or frequent. The code has made use of indention, making it clear where each new section starts.**e.g. # this code creates the conversation loop.**Whitespace is used to break the program text into conceptual blocks.*The student has:* tested and debugged the program to ensure that it works on a sample of expected cases

**For example (partial evidence)***The student has written functioning code that works on expected input cases There is evidence of expected cases that were used to test and debug the program to show that the program works when the user inputs data that is expected The student has written about some expected case data and has tested what happens when the program is run. Such testing may be observed and recorded by the teacher, presented in table form with minimal notes, recorded using a screen cast or described using examples.**The examples above are indicative samples only* | Develop an informed computer program. The student has:* documented the program with variable names and comments that describe code function and behaviour

**For example (partial evidence)***There are clear comments throughout the code that describe relevant functions or sections of code. The variable names clearly describe the data they hold.**e.g. # this function selects the appropriate sentence for the chatbot to say next. It will present the next response.**e.g. response = pick\_next\_sentence(word\_used)*The student has:* followed conventions of the chosen programming language

**For example (partial evidence)***For example, if the student is using Python: variable names use snake\_case rather than CamelCase; constants are ALL\_CAPS with underscores separating the words if required;**files and functions contain a docstring explaining the purpose of the program/function.**.**If the student is using Python, function definitions appear before the main section of code and the main section is all at bottom, not between the functions, thus making the program easier to read.*The student has:* tested and debugged the program in an organised way to ensure that it works on a sample of both expected and relevant boundary cases

**For example (partial evidence)***Evidence is presented of the tests planned and passed. These will show how the final program behaves when the user inputs data that is expected, and how the program behaves on boundary cases* *Such testing may be presented in table form, recorded using a screen cast or described using examples.**The program might not correctly handle unexpected data.**The examples above are indicative samples only* | Develop a refined computer program.The student has:* ensured that the program is a well-structured, logical response to the task

**For example (partial evidence)***The student’s final program consists of multiple functions/methods so that the program flow is clear and there is no unnecessary or redundant code. Functions/methods are well named so that they are self-documenting.**The code is clear, concise, and easily readable. The main program may be short and consists of multiple user defined functions which do most of the logic and processing.**Student code is laid out so that all the function definitions appear before main program.*The student has:* made the program flexible and robust

**For example (partial evidence)***The program is easily modifiable, so that it can be adapted to new contexts, for example discussing different topics, or using Māori rather than English.* *The program uses actions, conditions and control structures effectively and without unnecessary repetition.**The program uses constants, variables and derived values in place of literals. Constants, for example, appear at the top of the code where they can easily be seen and changed should those values change over time.**The program avoids the over-use of global variables.**The program correctly handles expected, boundary and invalid values.**They have used appropriate error handling techniques to ensure that program handles even unexpected cases.*The student has:* comprehensively tested and debugged the program

**For example (partial evidence)***The program checks the validity of input data and deals with expected, boundary and unexpected or invalid cases. All cases have been comprehensively tested including testing each condition within their code.**Testing has been done in a systematic way. Test cases have been well thought out and notes may have been made showing that the code works as expected for* ***all*** *use cases.* *The examples above are indicative samples only* |

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the achievement standard.

**Assessment schedule/Mahere Aromatawai: Digital Technologies & Hangarau Matihiko 91884 – Chatbot Charlie**

|  |  |  |
| --- | --- | --- |
| **Evidence/Judgements for Achievement/Paetae** | **Evidence/Judgements for Achievement with Merit/Kaiaka** | **Evidence/Judgements for Achievement with Excellence/Kairangi** |
| Use basic iterative processes to develop a digital outcome.The student:* plans a digital outcome to address a problem, need, opportunity or interest

**For example (partial evidence):***The student presents pseudocode showing the algorithmic structure of their program and describes the user interaction.*The student:* manages the development of the digital outcome by decomposing the digital outcome into smaller components

**For example (partial evidence):***They have decomposed their basic computer program into the components that need to be developed. Components could include data structures, functions, user interface.* The student:* trials components of the outcome in an iterative manner

**For example (partial evidence):***The student has used end users to trial the components as they are developed. Each component is trialled in an iterative manner until the final chatbot program is produced. This is visible within their development log.*The student:* tests that the program functions as intended

**For example (partial evidence):***The student tests the functionality of the program and that it can hold a conversation. This is evidenced within their development log.*The student:* describes relevant implications

**For example (partial evidence):***The student describes what cultural implications are, and why this is relevant to a chatbot; Similarly, the student describes what other implications are, and why this is relevant to their program.**The examples above are indicative samples only* | Use basic iterative processes to develop an informed digital outcome.The student:* uses information from testing and trialling to improve the outcome

**For example (partial evidence):***The student records what tests and trials have been performed, possibly using annotated screen shots or annotated video and shows how tests and trials have led to improvement.*The student:* trials multiple components and/or techniques and selects the most suitable

**For example (partial evidence):***They have trialled two different techniques for choosing the next response and selects an option that does not cause functionality issues.* The student:* addresses relevant implications

**For example (partial evidence):***They have addressed the fact that it is unethical to use copyrighted code through creating their own code for the chatbot.* *They also ensure that any conversation is socially and culturally acceptable.* *Usability and aesthetic considerations have been addressed through testing how the next response is presented to the user and thinking about different ways this information could be presented.* *The examples above are indicative samples only* | Use basic iterative processes to develop a refined digital outcome.The student:* applies information from the planning, testing and trialling of components to develop a high-quality outcome

**For example (partial evidence):***Evidence has been provided that their planning, testing and trialling has allowed them to develop all components iteratively and applied this information to produce a high-quality digital outcome.**Their outcome functions as intended and has no obvious errors in functionality or presentation of the information.**Evidence gained from trialling is thorough and organised testing has been integrated into the outcome in an ongoing manner to ensure the outcome is of high quality, including aesthetics, functionality and usability.**The student has clearly addressed any ethical issues that arose in the responses of the chatbot and has taken care that the chatbot conversation will not offend any user.* *The examples above are indicative samples only* |

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the Achievement Standard.

1. These achievement standards are derived from both *The New Zealand Curriculum* and *Te* *Marautanga o Aotearoa.* [↑](#footnote-ref-1)