Internal Assessment Resource

Digital Technologies Level 2

2023

This resource supports assessment against

[**AS91896**](https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91896.pdf) V1 - Use advanced programming techniques to develop a computer program (6credits)

[**AS91897**](https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91897.pdf)V1 - Use advanced processes to develop a digital technologies outcome (6 credits)

**Instructions:**

* Read the entire assessment task and assessment schedules for programming and processes
* Spend four hours in class and 2 hours out of class per week
* Open this document in Desktop version, do not save to any other location as it saves on SharePoint
* You must comply with the Occupational Health and Safety in Employment Act 1992, and their subsequent amendments.
* You must ensure you have not breached the Copyright Act
* Complete ALL tasks, attach your final py file with code commenting to Teams and Click ‘Turn In’

**Authenticity Statement:** This is an open book assessment, and you can go to various sources for guidance—for example, OneNote, CodeAvengers. Acknowledge sources of learning, including any help from teacher, friends, and family before or during this assessment. If work is non-authentic in any aspect, no credit will be given regardless of the quality. This will apply to **ALL** pupils involved in the misconduct. There will be no opportunities offered for further assessment in this standard for the current year.

Tick the box to acknowledge that you read and understood the Authenticity Statement

**Resubmission:** A **resubmission** may be offered when a student can gain an Achieved grade, if they correct minor errors or omissions in their work in a short period of time. There is no resubmission for Merit or Excellence grades. Therefore, it is essential to check assessment schedules for both standards regularly.

# Overview/Kupu Arataki

This assessment activity requires you to **plan**, **trial**, **test** and **develop** a computer program using advanced programming techniques. You will use a development process to help you make informed decisions throughout the coding, testing and trialing of your program and show ongoing testing and refinement to improve the functionality and quality of your program.

You will be assessed on how effectively you plan your development, decompose the outcome into smaller components, test and refine your program so that it is a high-quality response to the task (e.g. well-structured, logical, flexible, robust and comprehensively tested).

When planning and developing your program, you must ensure your program:

* uses variables storing at least two types of data (e.g. numeric, text, Boolean)
* uses **sequence** (commands), **selection** (conditionals) and **iteration** (loops) control structures
* takes input from a user, sensor(s), or other external source(s)
* produces output

**AND** includes **two or more** advanced programming techniques, such as planning and writing code that:

* defines and manipulates multidimensional data in collections (e.g. lists, dictionaries)
* creates methods, functions, or procedures that use parameters and return values
* responds to events generated by a graphical user interface (GUI)
* requires non-basic string manipulation, e.g. slice
* uses the functionality of additional non-core libraries.w3q1

# Tasks/Hei Mahi- Choose any one of the brief to create a computer program.

Brief 1: Café Menu

Brief 2: A quiz on any chosen subject.

# Task/Hei Mahi

## Brief1.

The school café wants you to create a ‘click and collect app’ for café orders. It is an app for students of age 13-18. Existing student users may log in or create an account to use the app. Then the app displays menu with prices for placing orders. Once user chooses items, the app displazys order details - items, quantity, price and total.

## Specifications

* Introduction to programme
* Offers choice to log in or create account
* Checks for correct username and password in index data (list)
* Checks for age for new users and saves username and password in a list
* Displays Café menu and asks for orders – items, quantity
* Displays invoice with details of items, quantity, price and total

Brief 2 – A Quiz on any chosen subject.

. A text based or GUI based menu interacts with an array of questions. The questions should be of multiple choice. A record of correct/incorrect answers is kept, and statistics gathered about the responses. The menu allows for users to skip and retry questions as well as resetting the whole quiz. Multiple arrays are used to hold questions - attempted and not-attempted. If working with GUI, make sure the GUI includes drop-down lists and at least one of radio buttons or text-boxes.

You may use the above ideas as starting points for your own project or you may come up with a unique idea of your own. Visit [Project ideas](onenote:https://mybdscschool.sharepoint.com/sites/3DIP_21/SiteAssets/3DIP_21%20Notebook/_Content%20Library/Scholarship.one#Project%20ideas&section-id={3B929117-8484-4FB6-9987-28526616E4B7}&page-id={CB389327-2F88-4E7A-BDE6-9745F28202FA}&end) ([Web view](https://mybdscschool.sharepoint.com/sites/3DIP_21/_layouts/OneNote.aspx?id=%2Fsites%2F3DIP_21%2FSiteAssets%2F3DIP_21%20Notebook&wd=target%28_Content%20Library%2FScholarship.one%7C3B929117-8484-4FB6-9987-28526616E4B7%2FProject%20ideas%7CCB389327-2F88-4E7A-BDE6-9745F28202FA%2F%29)) for more ideas. **If you cannot come up with a project within two lessons, a project may be given to you.** Your idea, and the context it works in (i.e. how it is unique to you) must be discussed and agreed upon by your teacher before you start the work outlined in the planning section, below.

Programming code should be set out clearly. Document the program with appropriate variable/module names and organised comments that describe code function and behaviour. Use appropriate variable/module names and follow conventions for your chosen programming language.

* Show comprehensive testing and debugging of the program. This should be carried out in an organised way to ensure that it works on expected cases and relevant boundary cases. (This could possibly be shown in your versioning, or through screencasts.)
* Ensure that the program is a well-structured, logical response to the task.
* Ensure that the program is flexible and robust.

Brief 3:

You can create any other computer program that addresses an issue with your provided solution.

## Steps for Processes and Programming

1. Decide on an appropriate planning methodology, and what project management and version control tools you will use to manage your program development. *(Log this in the* [*table below*](#_m333xcp1e3o0) *and in your* [*work journal*](#_9l4yhpkwnuz0)*.)*

2. Set up any necessary planning/project management tools. *(Log what you’ve done in your* [*work journal*](#_9l4yhpkwnuz0)*. Screenshot your progress in the project management system as you work each day.)*

3. Research any content for your program and use your chosen tools to plan out the development and structure of your program.

For example:

* An overview of how your program will work? *(Log this in your work journal.)*
* How will you incorporate important information about your topic for users? *((Log this in your work journal.)*
* Will the program use graphical elements or be text-based? *(Log this in your work journal.)*
* Explain what relevant implications are important to consider in the development of your program? *(Log this in the relevant implications section of this document.)*
* Decompose your program into the different components you need to incorporate into click and collect app (e.g. start-up screens/GUI interface, different levels, sub-programs/functions/methods). *(Log this in your work journal, eg. assign different tasks to Trello and screenshot into your work journal.)*

4. Throughout your development, you must trial multiple components. For example, this could include different ways to present the user interface, different ways of adding totals, etc. You should also think about the advanced programming techniques that will best make your program flexible and robust. Select the best components and/or advanced programming techniques to include in your final program, based on the results of your testing and trialling. *(Testing and trialling of components can be done in a separate repl.it project with public access and linked into your work journal. Alternatively, there is also a section below for this. )*

5. Use your selected version control tools/techniques to save successive versions of your code and keep evidence of how you created the program in an ongoing manner (e.g. screenshots showing your file structure with appropriately named versions/program components, including brief annotations of the changes made in each version). *(Log this with screenshots in your work journal. This can either be done with a git repository or by using a file naming system to keep track of versions.)*

6. Ensure your testing and trialling includes both expected cases and relevant boundary cases (e.g. what happens when the score/lives gets to a certain number). You may want to get other students, your family/whānau, and/or the Environmental club members to test your program at each stage and provide feedback to help you improve your final click and collect app program. Using others to test the program will help to ensure it is comprehensively tested for many different cases (including expected and relevant boundary cases). Note the improvements that could be made based on the testing and implement your changes. *(Log your thoughts on improvements in your work journal. Log input test results in the* [*input testing*](#_e3pbt8rqluzw) *section of this document.)*

7. Throughout the development of your program code, ensure that you document your program with appropriate variable/module names and comments that describe code function and behaviour. Follow the common conventions of your programming language (e.g. naming conventions or rules for program layout). ***Uncommented code will result in Not Achieved.***

8. Comprehensively test your final program to ensure that it functions correctly and is of high-quality (e.g. bug free, has a well-presented and easy-to-use interface, contains all the required information). (*Log your ongoing testing and refinements in your work journal.)*

***Note****: While input testing should be logged in the testing section of this document, the general testing of your program can also be recorded by making a brief screencast showing the outcome being comprehensively tested. If desired, you can take screenshots of your screencast and annotate them. This is often easier than trying to screenshot whilst testing where it is easy to ‘forget’ to screenshot a key part of the test. If you prefer, you can record a voice-over of the screencast to explain your testing and simply submit the screencast.*

9. Discuss how the information from planning, testing and trialling of the components of your click and collect app program assisted you to develop a high-quality outcome. This can be in the form of a screen-cast, document with annotated screenshots, online presentation or oral presentation to your teacher/class. *(Log this in the appropriate table below and in your work journal.)*

10. Show how your program has addressed the relevant implications. *(Complete this in the Relevant Implications section of this document.)*

# Task 1 Project and project requirements

* Start with a good introduction to your project :

Click and collect app BDSC café for students aged 13-18.

Purpose: The purpose of the program is to create a click to collect app for the school cafe, this solves the problem of students having to queue in line and taking orders. It will speed up the process if taking orders.

End Users: The end users are BDSC students.

Software program used: VS code & python.

|  |  |
| --- | --- |
| Name of app |  |
| Project management | Give a shared link to Trello. Add screen shots of project management to developmental logs. Your development logs in the Trello should show effective management of development of project with timelines and their completion. |
| Version control system | Save files as different version and submit them for your final submission! List the Versions of the program:  Github Link: |
| Flowchart/Pseudocode | Give a shared link to draw.io or similar. Add screen shots of plan in Task 5 |

# Task 2 Developmental Logs (Processes)

|  |  |
| --- | --- |
| Date | Log what you’ve done each day on the project. Alternatively, you may use any screen casting software to record your desktop and voice explaining what you would otherwise write in these tables. This is important even if you just want a grade of **Achieved**. **Merit** and **Excellence** require that you show how your project developed throughout including but not limited to use of design thinking steps, independent research, issues you encountered and how you overcame them, ongoing testing procedures. Provide clear evidence of ongoing evaluation, and feedback from self and potential users for improvement.  *Add rows as needed.* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Task 3 Testing Log

|  |
| --- |
| **Ongoing testing is collected as evidence for programming and processes, therefore, update this table(including date) during the development . This is different from Test cases. Your testing log must include description of component tested, screen shot of result, what are the errors and how you have fixed them.**  *Add rows as needed.* |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# Task 4 Test cases (Programming)

Test cases describe different scenarios that occur when the user inputs data into the system you build. There are three basic types – expected, boundary and invalid. To call it comprehensive testing, you must plan several test cases in the first four columns before the development and complete the fifth column before you hand in:

* Expected: These are values that you would expect the user to enter in normal circumstances. They fall within the expected range of input. For example: age = 5 or 18 or 50 or any number from 0-100
* Boundary: These are values that are just at, below and above the boundary values
* that are just on the edge of what you would expect the user to enter in normal circumstances. If the minimum age of the user is 13. Then age = 12 or 13 or 14 are boundary values.
* Exceptional (Invalid): These are values that you wouldn’t expect the user to enter in normal circumstances. They fall outside the expected range of input. For example: age = 150 or ten or -15 or 13.5 or 1 8.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User input (*A short explanation)* | Test Case  (*What would the user input?)* | Input Type  *(expected, boundary or invalid)* | Output  *(How would your system respond?)* | Actual Output  *(Screenshot)* | What are next steps? How to fix or improve |
| Enter username: | Eddie | expected | Program asks for password | Enter password: |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Task 5 Flowchart or Pseudo code of your programme

Add screenshots of draw.io for main routine and each function. They must be clear and readable. There must be several iterations of your flowchart or Pseudo code (show teacher a sample of your pseudo code for approval) during your development.

A flowchart and its explanation is needed for each version

Task 6 Explanation of relevant implications

For **achievement**, choose *at least* **three** implications from the list in OneNote. Define what is the relevant implication , describe why it relevant for this project and explain how you addressed those implications that relate to your project. For **merit/excellence** you will need to show evidence of at least five of these relevant implications.

|  |  |  |
| --- | --- | --- |
| Define what the implication is. | Describe why it is relevant | Explain with annotated screen shots of how you addressed this implication |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Task 7 Reflective summary

Present a summary of how the information from planning, project management and design thinking processes, relevant implication, and testing and trialing assisted you to develop a high-quality outcome. This may include annotated screen shots of Trello, programme and iterative process that guided your development process and helped you complete all components and manage your time effectively.

Task 7 Source of learning

Acknowledge sources of learning, including any help from teacher, friends, and family before or during this assessment.