**GRADE: N / A / M / E**

**Assessment schedule/Mahere Aromatawai: Digital Technologies & Hangarau Matihiko 91896 – Use advanced programming techniques to develop a computer program**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Evidence/Judgements for Achievement/Paetae** | **Evidence/Judgements for Achievement with Merit/Kaiaka** | **Evidence/Judgements for Achievement with Excellence/Kairangi** | | | |
| **Use advanced programming techniques to develop a computer program.** | **Use advanced programming techniques to develop an informed computer program.** | **Use advanced programming techniques to develop a refined computer program.** | | | |
| **Written code for a program that performs a specified task:**  ☐ Writing a working program to a specified task  ☐ The student has used Python to complete the project  ☐ Program generates a random receipt number | **Has at least one of the following advanced techniques:**  ☐ Program has a working image  ☐ Program generates a random receipt number and places it in a row | **Has included at least ONE of the following:**  ☐ Have a unique receipt number generator  ☐ Have enhanced stylistic features including images and/or icons | | | |
| **Comments**  ☐ The code is internally documented with comments  **Mandatory:**  ☐ Use variables storing at least two types of data (e.g. Numeric, text, Boolean, object)  ☐ Used a GUI library Tkinter  ☐ Use sequence, selection, and iteration control structures  ☐ Take input from a user save to a data file  **The program must have at least TWO of the following advanced techniques:**  ☐ Creating methods, functions, or procedures that use parameters and/or return values.  ☐ Responding to events generated by a GUI  ☐ Using of concatenation  ☐ Using additional non-core libraries  ☐ Have a main()  **Making the program flexible and robust, such as…**  ☐ Creating reusable functions that are found throughout the program  ☐ Conditions ☐ Control structures ☐ Methods  ☐ Has used two functions  ☐ Using constants, variables and derived values in place of literals  **Testing inputs with expected and actual outputs:**  ☐ Every item must be tested, and testing table is provided  ☐ Program works on expected input  *Program may crash on boundary or invalid input.*  ☐ Student has tested all expected cases to qualify the different conditions  ☐ A paragraph is provided explaining the testing process  **Debugging**  ☐ Fixed the program to ensure that it works on a sample set of expected cases. *Students have placed snips/copies of their fixes* | **Comments**  ☐ Every block of code has relevant/descriptive comments  ☐ Comments: Use a space after the #, first word capitalised, use a full sentence ending in a period, indent to the same level as the code it’s describing.  **Naming Conventions**  ☐ The names of the functions and variables are relevant and descriptive  ☐ The name of the variables and functions are in lower case  **Testing valid and invalid inputs with expected:**  ☐ Student has included a test table with all boundaries tested and outputs  ☐ Debugged the program in an organised way to ensure that it works on a sample of both expected and relevant boundary cases (e.g. Max/Min testing)  ☐ On unexpected inputs should generate an error message. Might not be as clear. | ☐ The program is a well-structured, easy to read, logical, detailed and understandable response to the task and inputs  ☐ Functions have been used to keep distinct tasks separate, and to avoid duplicate code  **GUI and the underlying code are kept separate and communicate via a well-defined interface:**  ☐ No console output  ☐ Use of a combo box/spin box  ☐ Formatted and clear entry boxes  ☐ Formatted and clear labels  ☐ Separate message boxes/or very clear instructional messages for valid and invalid inputs  ☐ Made the program flexible and robust – multiple use of methods, all conditions, boundaries  ☐ Program Reads/Writes to a curated .txt, .json, .xls file.  **Comprehensively tested and fixed the program:**  ☐ Program must be bug free  ☐ Every item needs to be comprehensively tested (snips are included)  **Comprehensively testing all:**  ☐ Unexpected inputs  ☐ Expected inputs  ☐ Boundary values  ☐ Detailed and instructional error message for alphanumeric and unexpected inputs (e.g. full stops, symbols, special characters, decimals, ASCII characters)  ☐ It has been structured so that making changes to it is easy. | | | |
|  | **GRADE** | **N** | **A** | **M** | **E** |

**Comments:**

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