**Final Submission**

On the due date, you are required to hand in the following for assessment:

* Evidence of your testing. This should be entered in the Word document called firstname-lastname-testing.docx.
* An electronic copy of all your versions. This can be different files or your commit history on GitHub.

**Programming Checklist**

To pass this assessment it is necessary for you to have met certain requirements

Before you hand in your assessment for marking ensure you have done the following **:**

|  |  |  |  |
| --- | --- | --- | --- |
| Program works as per specifications: |  |  |  |
| * It must ask for the course |  |  |  |
| * It must ask for the item(s) to be ordered |  |  |  |
| * It must ask for the quantity of each item |  |  |  |
| * It must limit the quantity of each item to a maximum of 5 |  |  |  |
| * It must calculate the total cost of the order |  |  |  |
| * It must display the order details |  |  |  |
| * It must enable the waiter to confirm or cancel the order |  |  |  |
| * Only items that are available are displayed |  |  |  |
| The program must be set out clearly and documented with comments and docstrings |  |  |  |
| The program must use variables of at least two different data types |  |  |  |
| Program includes variables and lists |  |  |  |
| Program must contain input and have output |  |  |  |
| Have conditional statements such as IF |  |  |  |
| Have at least one loop such as WHILE or FOR |  |  |  |
| It must have user-defined functions |  |  |  |
| Uses 2 complex skills:   * Uses classes and objects * Includes a working GUI using Tkinter * Reads to/from a file |  |  |  |
| *Video capture of program working is submitted (not required in 2020)* |  |  |  |
| The program has been tested for expected cases |  |  |  |
| Evidence is provided that the program has been developed iteratively |  |  |  |
| **Teacher guidance required / worked independently throughout** |  |  |  |
| Enables the user to add or delete items from the menu |  |  |  |
| Enables the user to update availability of items |  |  |  |
| The user is required to log in with an approved username before updating the menu |  |  |  |
| Have followed the style guide |  |  |  |
| Variables, function, list and class names accurately reflect their function and behaviour |  |  |  |
| Parameters are passed into functions |  |  |  |
| Uses well-chosen functions, classes, GUI elements and event handling mechanisms. |  |  |  |
| Have sufficient comments in the program to describe its function and what each section does |  |  |  |
| Program works with expected and relevant boundary cases |  |  |  |
| Evidence of organised testing, showing program works for expected and relevant boundary cases |  |  |  |
| Program must be well structured and be laid out logically |  |  |  |
| Logical decomposition of the task |  |  |  |
| Have functions where they should reasonably be used |  |  |  |
| Use variables, constants and derived variables effectively and robustly |  |  |  |
| The program must be concise – be efficient with the amount of code required |  |  |  |
| Have comments that explain what each step is doing and justify why |  |  |  |
| Program works for expected, relevant boundary and invalid inputs |  |  |  |
| Evidence is provided to demonstrate that testing has been comprehensive |  |  |  |
| GitHub commit history demonstrates well-organised development process |  |  |  |