

Coursework Assessment Matrix

Module Title	Programming for Data Science	Module Code	24COP504
Student ID		Mark/Grade	

	Fail	Borderline Fail	Pass	Merit	Distinction
<i>Program Structure: Logic</i> (approx 14%)	Program logic is fatally flawed, and program does not operate as required. Program fails to run.	A few problems with Logic. The program performs some of its tasks with runtime error.	Logic is sufficient for the program to perform most of its tasks with only the occasional runtime error.	Logic is sound. Basic Test code is provided at the bottom of each program that tests the main functions within the program file.	Logic is very good. Test code is comprehensive and provided at the bottom of each python program that tests all the functions within the module with exceptionally good data.
<i>Program Structure: Features</i> (approx 60%)	Program falls short of fulfilling the specification and cannot be considered usable.	Program implements all but one or two less important features of the specification and can be considered just usable.	Program operates according to minimum requirement. The program is fit for purpose.	The program exceeds the brief in user interface aspect of its functionality as described in the specification.	The program demonstrates exceptionally good use of GUI elements and sophisticated rules for identifying top artists.
<i>Maintainability: Style</i> (approx 20%)	Variable names inappropriate or unclear [e.g. use of TLA's (three letter acronyms) throughout].	Variable and function names are for the most part appropriate and structure is clear and readable. Indentation uses four spaces rather than tabs	Variable and function names are well thought out and used throughout. Global and local variables are used appropriately. Appropriate data structures are used (e.g. List, Array and Dictionary etc). Modules used as described in the specification.	Variable, function and module names are used in a way that means the code documents itself. Reusability of functions has been well considered in the program/module design.	Class types are used very well to improve the software quality and modules are used to improve the reusability.
<i>Maintainability: Docstring/markdown and comments</i> (approx 6%)	Comments or docstring extremely limited and/or little or no use to the programmer or end user.	Very important docstrings and comments are missing. Some parts of the code are over commented.	Sufficient comments, docstrings and markdown are included to make it possible to follow the program logic.	All functions and modules are provided with an appropriate docstring. There is little evidence of over commenting.	Comments are used judiciously throughout. Docstring provided are comprehensive and clear. Markdowns are diligently used to create sections

Other feedback:

Signed:

Notes:

1. The final grade will be allocated according to distribution of grades as indicated in the matrix above together with academic judgment (this is necessary as both grades Distinction and Fail span disproportionate percentages when compared to the other grade boundaries).
2. Typically, grades in the table should be considered cumulative – that is it is not possible to achieve a Merit without also demonstrating the outcomes in the boxes for a Pass first.
3. if the supplied programs do not run the project will be marked as a **fail** and be allocated a mark of **0**. Please ensure you test all your modules run before submitting your coursework.