Assignment Title:	AB1 – Python Project		**Deadline:**	04/03/2025 – 23:59
Course:	EAS Hybrid Course for Secondary Education		**Unit:**	Unit 1
Learner:		**Marker:**	**Moderator:**	
	Pass (High) 100 – 80 %	**Pass (Mid)** 79 – 60 %	**Pass (Low)** 59 – 40 %	**Fail** 39 – 0 %
Log In 3 marks	Excellent implementation. Works for multiple users and is very efficient. May update output based on the number of failed attempts	Good log in implementation. Works for multiple users and is reasonably efficient.	Passable log in implementation. May only work for one user. Code has various inefficiencies.	Functionality broken, missing or very poor
Menu 4 marks	Excellent menu that is highly attractive and intuitive to navigate. Makes use of submenus to provide a clean interface. Flawless system integration.	Good menu implementation that is attractive and easy to read and navigate. System integration is good.	Acceptable menu implementation that is straightforward to read and navigate. Passable system integration.	Functionality broken, missing or very poor. Difficult to read and navigate. Poorly integrated system.
Add & Remove Learner 6 marks	Learner data is stored using a suitable data structure. Learners can be added and removed according to the stated requirements with a high level of efficiency. Learner data can be externally saved and loaded	Learner data is stored using a suitable data structure. Learners can be added and removed according to the stated requirements with reasonable efficiency	Learner data is stored within the system. Learners can be added and removed from the system but may not require all necessary information, or requires more information than stated in the requirements	Learner data not stored, or missing data. Unable to add or remove learners
View All Learners 3 marks	All learners are printed to the screen with very clear order and grouping that is easy to read and aesthetically	All learners are printed to the screen with a clear order that is easy to read. Implementation is reasonably efficient	All learners are printed to the screen. May be unordered, unattractive, or difficult to read. Implementation may be inefficient	Functionality broken, missing or very poor

	pleasing. Implementation is highly efficient.			
Add, Remove & Change Assessment 3 marks	Assessment information can be added, removed and changed for each learner according to the stated requirements. Implementations are highly efficient.	Assessment information can be added, removed and changed for each learner according to the stated requirements. Implementations are reasonably efficient.	Assessment information can be added, removed and changed for each learner. May be possible to do so without providing all necessary information. Implementations may be inefficient.	Functionality broken, missing or very poor
View Learner Assessments 3 marks	Assessment data for a specified learner is printed to the screen with very clear order that is easy to read and aesthetically pleasing. Implementation is highly efficient.	Assessment data for a specified learner is printed to the screen in line with the requirements. Output is ordered and easy to read. Implementation is reasonably efficient.	Learner assessment data is printed to the screen. May be unordered, unattractive, or difficult to read. Implementation may be inefficient.	Functionality broken, missing or very poor
Additional Functionality 5 marks	Sweeping enhancements have been made that greatly improves the program and showcases an in-depth understanding of a python feature/library not covered by the assignment	Considerable additional functionality has been added that improves the program and showcases python knowledge not covered by the assignment	Substantial additional functionality has been added that is effective in improving the program	None or very minor additional functionality, or a poor implementation that makes the application worse.
Code Quality 5 marks	Excellent code quality. Consistent with the standards of best practise to a professional level. A second programmer could take over with little time to familiarise themself	Good code quality. Consistent with the standards of best practise with few mistakes. A second programmer could take over with some time to familiarise themself	Acceptable code quality. Mostly consistent with the standards of best practise but may have some errors (e.g. inconsistent variable names, magic numbers, too few/many comments)	Poor quality, inefficient code that ignores the standards for best practice

Modularity 3 marks	Effective use of separate modules to break up the code in a sensible way. Individual modules may contain various functions if appropriate. Loose coupling, high cohesion.	Good use of functions to break up the code. Consistently applied across the application. Loose coupling, high cohesion.	Functions have been used but there may be too many/too few. Implementation is inconsistent. Middling coupling and cohesion.	None or limited attempt at program modularity. Very few functions that are poorly implemented, else the entire project is one continuous program. Tight coupling, low cohesion.
Knowledge & Understanding **5 marks**	Excellent python knowledge demonstrated. Choices made for implementation are most suitable throughout. Comments convey an excellent understanding of the potentially complex structures/processes/features. used	Good knowledge of python demonstrated. Choices made for implementation are suitable throughout. Comments convey a good understanding of the structures/process/features used	Passable knowledge of python demonstrated. Choices made for implementation are mostly suitable. Comments convey a reasonable understanding of the structures/processes/features used	Poor knowledge of python demonstrated. Comments do not convey an understanding of the structures/processes/features used