Topic	Amplification	Teacher Guidance
Design	design and document the input and output facilities required to produce an effective user interface	Candidates should give due consideration to effective GUI usage.
	design and document all required data structures	Candidates should be able to design arrays, tuples, data dictionaries and lists and know which is most appropriate to use.
	using pseudo code, design and document the following routines: • validation and verification • data handling and processing • authentication	Candidates must be familiar with the use of pseudo code and will be penalised in the examination for using Python in its place. Candidates should know how to design and when it is appropriate to use various validation and verification techniques including: • range checks • presence checks • lookup table • check digit • length checks • format checks • double entry. Candidates should be able to design authentication techniques to ensure access to authorised users only.
Implementation	design, write, test and refine Python 3 code using the following skills: • create new and extend existing functions or methods • create new and edit existing objects • create new and extend existing Python 3 libraries • use variables (labels), operators, inputs, outputs and assignment • use a variety of data types, including integer, Boolean, real, character and string • use programming constructs to control the flow of a program, including:	The version of Python 3 will be specified on the initial task which will be carried forward on to the associated examination. Candidates will be supplied with a working unannotated program based on the initial task. They will be required to annotate, add and amend functionality to this program during the examination. Candidates should be familiar with basic Python library functions such as TkInter. Candidates should document code explaining its functionality.

	• iteration (condition and counter controlled loops) • selection • sequence • use basic file handling, including: • open a file • read from a file into a variable • read from a file into an array • write to a file • write to a file from an array • close a file • create new and extend data structures and fixed length records to store data • use string manipulation • create new and extend lists, tuples and dictionaries (arrays) • use slicing • use mathematical and logical operations • create new and modify existing intuitive graphical user interfaces using the Python 3 built in libraries including: • text boxes • buttons • forms • create and modify data validation and verification routines • create and modify authentication management routines • create code for the solution that is self-documenting and uses meaningful identifiers • use a programming style that is consistent, including indentation and appropriate use of white space • use subroutines with well-defined interfaces	

	• annotate code so that it is accessible to a competent third party • explain the solution or changes made to a solution.	
Testing	• design and document an effective testing strategy that will ensure that the final solution meets the requirements • describe how the outcomes of the testing process can be used to inform further development of the solution • design test data to include examples of typical, extreme and erroneous content • implement a test plan using typical, extreme and erroneous data where appropriate • present test outcomes with detailed and informed commentaries • demonstrate testing and refinement of code during development or in response to change in the requirements provided • explain the outcome of testing using given data or data that the candidate has designed.	Candidates should test their programs to ensure that they function correctly and produce the expected results. When inputting an integer between 1 and 32768: • Typical data could be 16184 • Extreme data could be 1 or 32768 • Erroneous data could be A or -3 or 3.14 Candidates should be able to justify their data choices.
Refinement	• demonstrate understanding of the requirements and any changes required by revised requirements • demonstrate understanding of changes by presenting evidence of developmental changes • demonstrate testing and refinement of code to improve efficiency	Candidates will be required to modify and test existing program code according to a new requirement. Candidates will be required to evaluate any changes they have made using the correct technical terminology.