READABILITY (/2)			
	Excellent (0.5)	Good (0.25)	Poor (0)
Program structure	Code structure highlights logical blocks and is easy to understand. Constants clarify code meaning.	Code structure corresponds to some logical intent and not make code too difficult to read.	Code structure makes the code difficult to read.
Identifier names	All variable and function names are clear and informative, increasing readability of the code.	Most identifier names are informative, aiding code readability to some extent.	Most identifier names are not clear or informative, detracting from code readability.
Repeated code	Repeated code has been avoided.	Some code is repeated unnecessarily.	Large amounts of code are repeated unnecessarily.
Variable scope	Only local variables have been used in functions; globals have been avoided.		Globals have been used.
SIMPLICITY (/1)			
	Excellent (0.5)	Good (0.25)	Poor (0)
Algorithmic logic	Code is simple, appropriately brief, and has no logical errors.	Code is not too complex or has minor logical errors.	Code is overly complex or has significant errors.
Control structures	Control structures are well used to implement expected logic.	A few control structures are a little convoluted.	Many control structures are used in a convoluted manner (eg. unnecessary nesting, multiple looping)
DOCUMENTATION (/2)			
	Excellent (1)	Good (0.5)	Poor (0)
All modules, classes, methods and functions have informative docstring comments	All docstrings are accurate, com-plete & unambiguous descriptions of how item is to be used. All parameters and return types, and expected values, are described clearly.	Almost all docstrings are accurate and reasonably clear descriptions of how item is to be used. Almost all parameters and return types are described clearly.	Some docstrings are inaccurate or unclear, or there are some items without docstrings. Some parameters and return types, or expected values, are not clear.
		Almost all comments provide useful	Some comments are irrelevant, not

Comments are clear and concise, without excessive or extraneous text		information that clarifies the intent of the code, making it easier to understand. Comments rarely repeat logic already clear in code.	providing any detail beyond what is obvious in the code. Comment style, or excessive length, obscures some program logic.
Significant blocks of program logic are clearly explained by comments		Important or complex blocks of logic (e.g. significant loops or conditionals) are clearly explained and summarised (i.e. stating a loop iterates over a list is not a useful explanation or summary).	Some important or complex blocks of logic are explained poorly, or are not summarised. Some unimportant blocks of logic are described in too much detail.