

Thomas I Ques 2 Assign 3

ASO2	Level 4	Level 3	Level 2	Level 1
communication	colouring to add to understanding, input one colour/output another	Flowchart - organized, proper spacing, correct symbols, proper language Comments - used succinct, (block OK), not sparse, block modifications, start date, programmer no problem description, Notes on anything special or unusual, citations Readability - code is easy to follow, enough white space	Poor spacing, some incorrect symbols, typing errors.	poor organizations, many symbols used incorrectly for the concept attempting to convey, arrows in the wrong section or incorrect pointer.
	Excellent User instructions that are clear and can be accessed at any time or Additional options, like premature quit.	user interface, clear language, no grammar or spelling errors I/O either CSV or txt - Open and closing, proper reading of all elements in the file and placing in an array/list/class IPO - proper chart (headings), steps, proper terminology	Adequate	Any two or more details missing or Citations/borrowed work not acknowledged OR more than 25% of code borrowed
Knowledge Thinking & inquiry	With open or similar more advanced coding Description at: top to clarify		2 or 3 grammar or spelling errors I/O opens and closes properly but may be inefficient OR may be skipping certain lines/words	User interface not clear, poor language and or spelling and grammar errors hard coded words and forgot to close files missing information in columns or steps missing
Thinking & inquiry	Screen sizes to user systems. Make it interactive. Colour or design helps user know what stage they are at.	Graphics are appropriate and professional looking for Hangman - word list appropriate components - Standard screen size 1280 x 1024 Trace statements - variable name, steps and changes (Table/chart) All functions (including user) are efficient and work as expected	Small graphics features not intuitive Variable names don't match or missed a step Functions work, but confusing or inefficient	No graphics, just console that user would not necessarily have. Lost count on number of guesses more confusing, complicate the existing error, or do not recognize it Hard coded where a user defined function or built in would suffice
Application	Could be class based Highly efficient, best possible, perhaps found a new function or way to complete the task	Code efficiency - nested if's better into function or class	Nested if's or lack of proper loops - does not take advantage of built in functions	No functions or classes

NO IPO, NO Trace ? Really needed interface to be all pygame or all turtle for top marks. Regular user would not have a console. Otherwise, good work.