Ben   Ques 4 Assign 3   Turtle Card Class

A3Q4	Level 4	Level 3	Level 2	Level 1
	Nothing assumed, catching errors, anticipating users needs - beyond	Comments - useful sincinct, (block OK), not sparse, block at top (modifications, start date, programmer name, problem description, Notes on anything special or unusual, citations)	lacking explanation, inline coments, or hard to read, or spelling mistakes on output or Progammer missing or modifications	Any two or more details missing or Citations/borrowed work not acknowledged OR more than 25% of code borrowed
communication	Level 3 Excellent	Readability - code is easy to follow, enough white space	Adequate	Needs improvement
	User instructions that are clear and can be accessed at any time or Aditional options, like premature quit.	user interface, clear language, no grammar or spelling errors	2 or 3 grammar or spelling errors	User interface not clear, poor language and or spelling and grammar errors
Knowledge	With open or similar more advanced coding - not required or not taken in class OR new or better implementataion of class work	Proper class usages and storing of data	Considerable usages and storing of data, few minor errors	hard coded words and forgot to close files
Thinking & inquiry	Description at top to clarify	IPO - proper chart (headings), steps, proper terminology	Unclear steps or incorrect terminology	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 [redacted] 21 - word list appropriate components - Standard screen size 1280 x 1024	Small graphics, features not intuitive.	No graphics, just console that user would not necessarily have. Lost count on number of guesses.
		Trace statements - variable name, steps and changes (Table/chart)	Variable names don't match or missed a step	more confusing, complicate the existing error, or do not recognize it.
Application	Could be class based	All functions (including user) are efficient and work as expected	Functions work, but confusing or inefficient	Hard coded where a user defined function or built in would suffice
	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
Flowchart/IPO/Tr ace Table pplication)	colouring to add to understanding, input one colour/output another	Flowchart - organized, proper spacing, correct symbols, proper language	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.

Calls + uses Orig Class ✓ - IF Calls own class - (% of orig. in your class) = 10% ?
%, of Orig. class used = 10%. Runs well.