

A303	Level 4	Level 3	Level 2	Level 1
communication		Comments - useful succinct, (block OK), not sparse, block at top (modifications, start date, programmer name, problem description, Notes on anything special or unusual, citations) Readability - code is easy to follow, enough white space user interface, clear language, no grammar or spelling errors Effectively used classes and inheritance	lacking explanation, inline comments, or hard to read, or spelling mistakes on output or Programmer missing or modifications Adequate 2 or 3 grammar or spelling errors used but not to best advantage or created cumbersome usage	Any two or more details missing or Citations/borrowed work not acknowledged OR more than 25% of code borrowed Needs improvement User interface not clear, poor language and or spelling and grammar errors hard coded words and or excessive repetition Poor organization, many symbols used incorrectly for the concept attempting to convey, arrows in the wrong section or incorrect pointer. missing information in columns or steps missing
Knowledge	Excellent User instructions that are clear and can be accessed at any time or Additional options, like premature quit. overloading or similar more advanced coding	Flowchart - organized, proper spacing, correct symbols, proper language IPO - proper chart (headings), steps, proper terminology	Poor spacing, some incorrect symbols, typing errors. Unclear steps or incorrect terminology	
Knowledge Thinking & Inquiry	colouring to add to understanding, input one colour/output another Description at top to clarify	Graphics are appropriate and professional looking for Yatze language - word list-appropriate components - Standard screen size 1280 x 1024 Trace statements - variable name, steps and changes (Table/chart)	Small graphics, features not intuitive Variable names don't match or missed a step	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.
Thinking & Inquiry	Screen sizes to user systems. Make it interactive. Colour or design helps user know what stage they are at.			
Application	Could be class based Highly efficient, best possible, perhaps found a new function or way to complete the task	All functions (including user) are efficient and work as expected Code efficiency - nested if's better into function or class. Flowchart - organized, proper spacing, correct symbols, proper language	Functions work, but confusing or inefficient Nested if's or lack of proper loops - does not take advantage of built in functions Poor spacing, some incorrect symbols, typing errors.	Hard coded where a user defined function or built in would suffice No functions or classes poor organizations, many symbols used incorrectly for the concept attempting to convey, arrows in the wrong section or incorrect pointer.
Flowchart/IPO/Trace Table (modification)	colouring to add to understanding, input one colour/output another			

90 MSDie Inhibit 30% - 40% - 50% - 60% - 70% - 80% - 90% - 100%
 1-100 Colour - ✓ Trace - ✓ Version - No Date - No IPO ✓
 Block @ Top Cannot reach last die - diff screen.
 One Round / Game PyGame