Card Game Rubric 2023 – Alpha Test

Coder Name: _____

Game:		Tester:	
# Methods:		# Cards:		Lines of Code:		Mark:	

Planning (Thinking)

Checkmarks			Level 1	Level 2	Level 3	Level 4	Level 4+
❑ April 12: Screen Up ❑ April 12: Card resized ❑ April 12: Opening Screen ❑ April 17: Baby Object ❑ April 18: Stack	❑ April 18: Shuffle ❑ April 18: Deals a card ❑ April 19: Instructions ❑ April 22: Score ❑ April 22: Alpha	❑ _____ early submissions	Under 5	5	6	8	10

User Interface (Knowledge, Communication)

Level 1	Level 2	Level 3	Level 4	Level 4+
❑ Game screen displays cards ❑ Score displayed	❑ Background colour, button colours, text colours changed. ❑ Instruction page ❑ Splash screen	❑ Instruction screen has picture of game. ❑ Splash screen is appealing, has game name and coder name.	❑ Consistent image choice ❑ Colour scheme used. ❑ Screens have unity: same colours/ theme/ image type used on each.	❑ Visually appealing ❑ Complex & detailed instructions ❑ Multiple pictures on instructions page.
❑ Title comments filled in	❑ Other comments in template filled in	❑ Comments added to global variables ❑ No reference to donuts, d or box ANYWHERE	❑ Comments added at start of every method ❑ Comments in Baby Object + Stack	❑ Outstanding Comments throughout ❑ Comments before major ifs

Functionality (Application, Thinking)

Level 1	Level 2	Level 3	Level 4	Level 4+
❑ Score can increase ❑ Code Runs ❑ A card is displayed	Baby Object: ❑ Mutators ❑ Accesors ❑ toString ❑ compareTo ❑ equals	❑ 15 cards ❑ Cards can change on screen ❑ Baby Object used	❑ Can reset ❑ Stack used to deal cards ❑ Easier complexity	❑ Good Parameters and Return Types. ❑ Higher levels of complexity added independently. **OR** more Complex game

More on Functionality for Level 4 & 4+ (Application & Thinking)

* You don't need all of these to have a Level 4+. That's crazy. I'm giving you some ideas.

* An easier game can still score a Level 4+. Just choose more from the first rubric.

* A challenging game needs to choose less from the first rubric. Their game already has a lot of complexity.

Add New Java Widgets & Functionality

Easier	More Challenging	Difficult	Very Difficult
❑ JTextArea ❑ OptionPanes ❑ JPasswordField ❑ Sound	❑ JPanels ❑ JRadioButton ❑ JCheckbox ❑ JSlider ❑ JProgressBar ❑ Complex OptionPanes ❑ 30 cards	❑ Allow Entry of Name, Update on Game Screen ❑ Use of String functions (charAt, indexOf, compareTo ❑ Additional object (dice, second card) ❑ Additional Stack ❑ 40 cards ❑ Own cards created	❑ Read/Write File to save/open game ❑ Checks answers ❑ Dictionary used to check answers ❑ Outstanding use of arrays ❑ Own method with parameters or return types ❑ More challenging games

More Complex Game:

Easier	More Challenging	Difficult	Very Difficult
Cards + Score ❑ Up A Bit, Hit List, Think N Sync	Multiple Objects ❑ Joe Name It	Checks Answer ❑ PDQ, Last Letter	Tricky Check Answer ❑ Adsumudi, Keep 100

Checklist

Curriculum	Checklist		
A4. Instructions/User Help	*Splash:* ❑ Name of game ❑ Name of coder ❑ Space filled ❑ Picture ❑ Button colour/font edit ❑ Title colour/font ❑ Background colour edit	*Instructions:* ❑ Picture of game ❑ More than one game pic, with arrows ❑ Detailed explanation ❑ Button colour/font changed ❑ Title colour/font edit ❑ Background colour edit. ❑ Consistent	*Game:* ❑ Reset button ❑ score buttons ❑ play again ❑ can move through cards ❑ Button colour/font changed ❑ Title colour/font edit ❑ Background colour ❑ Consistent
A4. Code Maintenance	❑ 3 Title comments in each class ❑ Game Runner: Comments before every method ❑ Game Runner: Comments before major loops and ifs ❑ Card: fully commented ❑ Stack: fully commented ❑ Comments continue right to the end ❑ No huge sections of commented out code ❑ No huge sections of random blanks ❑ Code is indented properly (just use the menu option!!)		

A2. Modular Programs/ C1. OOP & Modular Design	❑ own card class:. . . . .
	    ❑ getText:. . . . .
	    ❑ setText:. . . . .
	    ❑ compareTo:. . . . .
	    ❑ equals:. . . . .
	    ❑ toString:. . . . .
	
	❑ second class: :. . . . .
	❑ stack - for own class:. . . . .
	❑ second stack: :. . . . .
	❑ mutators, accessors created
	
	❑ organization - no methods over 30 lines
	❑ organization - own voids created:. . . . .
	❑ abstraction - no mention of donuts or d in card class
	❑ abstraction - no mention of donuts or d in game runner class
	❑ abstraction - no mention of "box" in runner class
	
	❑ abstraction - no mention of donuts or d in Stack class
	❑ extensibility - method with parameters/return type:. . . . .
	❑ extensibility - own method with parameters:. . . . .
	❑ extensibility - own method with return type:. . . . .
	❑ reusability - called own method at least twice:.		

D3. Emerging Tech/ D4. Exploring Computer Science	❑ OptionPane (Pop up) Line:. . . . . ❑ Complex OptionPanes Line:. . . . . ❑ Background Sound: Line:. . . . . ❑ Sound effects: Line:. . . . . ❑ Progress Bar: Line:. . . . . ❑ Pull Down/Combo Box: Line:. . . . . ❑ Radio Buttons: Line:. . . . . ❑ TextArea: Line:. . . . . ❑ Password Field: Line:. . . . . . . ❑ charAt:. . . . . ❑ indexOf:. . . . . ❑ toUpperCase:. . . . . ❑ substring:.		