ECE175--Fall 2018
Grading Rubric for Final Project

	Criterion	Maximum Points	Exemplary (100%)	Proficient (75%)	Marginal (25%)	Unacceptable (0%)
	Requirements					
	Intermediate Grading for April 17th					
1	Deck creation	/5	Deck is implemented as a linked list. The list is properly initialized and can be traversed. All cards are present in the deck	Deck is implemented as a linked list. However the implementation is incomplete, not all cards are present, or the list is not probably traversed	Some skeleton of a list is available, but the code does not properly compile	The deck is not implemented as a list.
2	Deck Printing	/5	Deck is correctly printed from the list	Deck is printed but the card format does not appear properly, making difficult to play the game	The print function does not work properly but some skeleton of the function is present	No printing function implented.
3	Deck suffling	/5	Deck is properly shuffled to a random deck	Deck is partially shuffled, but the deck does not appear sufficiently random	Deck shuffling breaks upon execution	Deck is not shuffled
4	Pseudo-code and code skeleton	/5	A pseudo-code is presented that accounts for all the game mechanics. Code is sufficiently modular and function prototypes are prototyped	A pseudo-code is presented, but several game functionalities are not accounted for. The code is not modular and only a few functions are prototyped	The pseudo-code is scarce and lacks essential details. Hadly any functions are prototyped.	No pseudo-code or functions are defined.
	Final Project Grading					
1	Deck creation	/5	Deck is implemented as a linked list. The list is properly initialized and can be traversed. All cards are present in the deck	Deck is implemented as a linked list. However the implementation is incomplete, not all cards are present, or the list is not probably traversed	Some skeleton of a list is available, but the code does not properly compile	The deck is not implemented as a list.
2	Deck suffling	/5	Deck is properly shuffled to a random deck	Deck is partially shuffled, but the deck does not appear sufficiently random	Deck shuffling breaks upon execution	Deck is not shuffled
3	Card Dealing	/5	Cards are dealt in the proper order.	The right number of cards is dealt, but not in a proper order	Cards are not dealt correctly	Card dealing is not implemented
4	Players' hands	/5	Players hands are impemented as a linked list. Memory is dynamically allocated with every card draw and freed after one card is played.	Players' hands are implemented as a dynamic list, but memory is not properly handled	Players hands are not implemented as a dynamic list	Players hands are not implemented

5	Drawing Pile	/5	The drawing deck is impemented as a linked list. Memory is dynamically freed as cards as drawn from the pile.	The draw pile is implemented as a dynamic list, but memory is not properly handled	Draw pile is not implemented as a dynamic list	Draw pile is not implemented
6	Discard pile	/5	The discard pile is impemented as a linked list. Memory is dynamically allocated and freed as cards are added to the discard pile. The pile is shuffled if the game has not ended by the time the draw deck is exhausted	The discard pile is implement as a linked list, but memory is not properly handled or shuffled when the draw cards are exhausted	Discard pile is not implemented as a dynamic list	There is no discard pile implemented
7	Game rules	/15	All game rules are followed	Most game rules are followed	Most game rules are not followed	Game is not functional
8	Game Interface	/10	The interface is inuitive. The user is able to play the game with minimal isntructions. Transitions from round to round are clear	The interface is intuitive for the most part. Some difficulty in understanding the game navigation.	The interface is counter-intuitive. Navigation options are not clearly stated. Interface limitations prevent proper game functionality	The interface is very basic and does not allow transitions between game rounds.
Program Design						
9	Code modularity	/10	The code is logically divided to several functions that implement important functionality (deck shuffling, dealer hand, player hand, hand tallying, list creation, list deletion, etc.)	The code is modular but further simplification could have been attempted	The code only has a few functions. Most functionalities are integrated within the main function	Code is not modular
10	Code documentation	/5	The code is properly documented. The input/output and goal of every fucntion is adequately described. Comments are provided for various parts of the code	The code is partially documented	The code is only cursorily documented	No documentation is provided
11	Compilation	/10	Code succesfully compiles without errors or warnings. The code does not hang while in execution	The code succesfully compiles, but some conditions may make it hang	N/A	Code does not compile
Total		**/100**				
Extra Credit						
12	Player Automation	/5	One of the players is played by the computer			

13	Multiple Players	/5	User can choose between 2 and 10 players			
14	Progressive Uno	/5	The progressive Uno rules are implemented			
15	Seven-O	/5	The seven-O Uno rules are implemented			
Total		**/20**				