Black Jack Rubric

Criteria	1 – Below	2 – Approaching	3 - Meeting	4 - Exceeding
Single Responsibility Principle	· Classes are extremely entangled · Multiple classes with duplicative behavior are present · Code is highly repetitive	· Most objects are disentangled, however game logic still presents some entanglement · No repetitive classes are present	· Classes are disentangled and reusable	· Game logic is contained within multiple specific worker methods and initiated by a coordinator method
Encapsulation	· Class field access modifiers are left public or default · Constructors are not declared public · Classes are not declared public · Methods are left at default access	· Some fields are made private, others are left default or public · Private methods are accessed via 'getters and setters'	· All class fields are private and accessed only through 'getters and setters' · Constructors, classes and methods are declared public	· Helper methods are kept private as they have no need to be accessed outside of the class
Object Oriented Programming	· Code is largely procedural · Existing classes do not follow Object Oriented principles · Local variables and collections are assigned to objects	· Classes exist which adhere to the Single Responsibility principle · Some procedural code exists, but is contained to bloated class methods	· Classes all adhere to Single Responsibility · Objects are instantiated and passed to other objects for modification/use · Objects are used to store and manipulate data, rather than local collections · Procedural programming is absent	· Polymorphism is utilized to represent like objects
Java Language	· Methods or variables are confusingly named or contain inappropriate language · Iteration over arrays	· ArrayList is used for collections of objects within classes · Method parameters are well named	· Methods are well named · For each loop is correctly utilized when iterating over arrays and ArrayLists · Default constructors are	· Code is well commented · Method and variable names create near human readable code

	excludes the use of the foreach loop, always defaulting to traditional for loops · Method parameters are confusingly named · Overwritten default constructors are not replaced · Class names are plural	· Class names are singular	replaced when overridden	
Black Jack	· The program does not compile · When run the game cannot determine a winner · Individual card values and rank are not relayed to the player	· The player can choose to hit or stay · If the dealer busts, the player wins · If the player busts, they lose · Ace value logic is complete · The deck of cards tracks which cards are available and is reshuffled when needed	· Players can place bets, bet value is tracked, when a player runs out of money they must either buy more chips or leave the game · Black Jack game logic is fully functional, players can both win and lose according to the rules of Blackjack	· Players can split hands · Multiple players can be in the same game · The number of decks used in a game can be selected