IT 179

Spring, 2022

**Program 5**

**Linked Lists, Stacks and Queues**

In this program you are designing a simulation of a simple card game. **(sample provided at the end of the document)**

The game uses only the numbered cards of the regular playing cards deck. (i.e.: 1 through 10 of hearts, spades, clubs, and diamonds)

The game can be played by a minimum of 2 players and up to 5 players. Each player will start with 4 cards and the goal is to collect 4 cards of the same type (for example all your cards are hearts) or have the total of the cards equal to 20.

The game will start by shuffling the cards and handing each player 4 cards. If one of the players has all their cards of the same type or the total of the numbers is equal to 20 that player wins.

If that’s not the case, the first player starts by drawing a card from the remaining cards which should be placed in a stack face down. After drawing that card, if they have 4 cards of the same type or if 4 of the cards, they are holding are adding up to 20 they win. Otherwise, they will have to throw a card to the floor face up. The next player will look at the card on the floor. If picking up that card will let them win, they will pick that card and win the game. Otherwise they will pick a card from the faced down stack of cards. Again, if after drawing that card they have 4 cards of the same type or if 4 of the cards, they are holding are adding up to 20 they win. If not, they will throw a card to the floor -face up- on top of the other faced up card(s).

The game will keep going until there is a winner or no more cards are left on the faced down stack of cards.

Classes:

1. A Game class which will include:

* A linked list to store the unshuffled cards
* A stack of the faced down cards
* A stack of the faced-up cards that are thrown to the floor
* A queue of the players
* A shuffle method
  + Randomly adds the cards from the linked list to the faced down stack
* Constructors:
  + Takes the player names
  + Calls the shuffle method
  + Create the player objects
* A method to simulate the game

1. A Player class which will include:

* A data structure to store the cards a player is holding.
* A constructor that sets the player name and takes the initial 4 cards.
* A method that will decide whether to draw a card from the stack or the floor
  + You should only draw from the floor if that card makes the player win.
* A method that will draw a card (either from the floor or the faced down stack).
  + This method should let us know if the player has winning cards after drawing
* A method to throw a card to the floor.
  + Should throw based on the card types (see example below)

1. Main class:

* Will have the main method
* Controls the flow of execution
* Takes input to start the simulation
* Create the game and start the simulation

**Notes**

* Running the game will result a simulation of the game (see below images) and output the cards of each player and the card on top of the floor stack at the end of each player turn.
* You can format your output in any way you want. You can add any more variables or methods that you need as long as you have the above methods and variables. You can use already available data structures from the java API or implement your own.
* **See the below sample to help you visualize the game.**
* Good program design is expected. Consider using private methods to break up longer methods into smaller pieces of functionality. Make sure everything has appropriate access control (private or public).
* Good commenting is expected with correct Javadoc style comments.
* Grading criteria:
  + MainClass(20%)
  + Game Class (40%)
  + Player class (40%)
* Deductions
* Late (-10% per calendar day)
* (-40) Syntax Errors
* (-30) Runtime Errors
* (-10) Style and Organization
* Reading material related to the assignment:
  + From Data Structures (IT179 book):
    - Chapter 2
    - Chapter 4

**Submission**

Zip your .java files and submit the .zip file to the Program 4 assignment on ReggieNet.

* + Email submissions will not be accepted
  + Follow the late policy in the syllabus
  + Corrupt files, empty files, invalid format files will result a zero

**Sample Game in the next page**

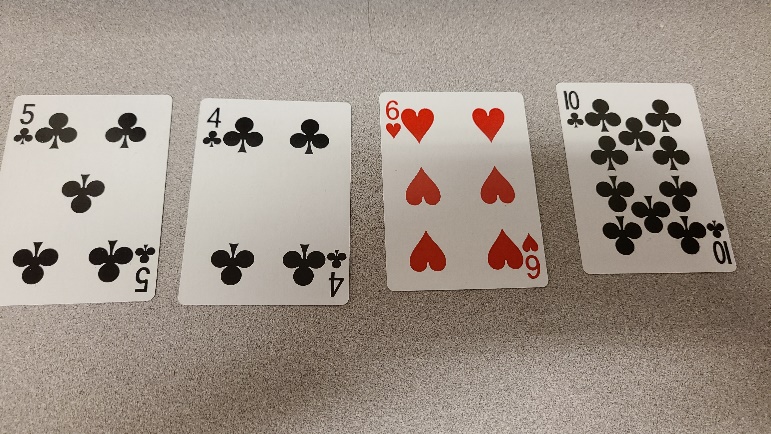
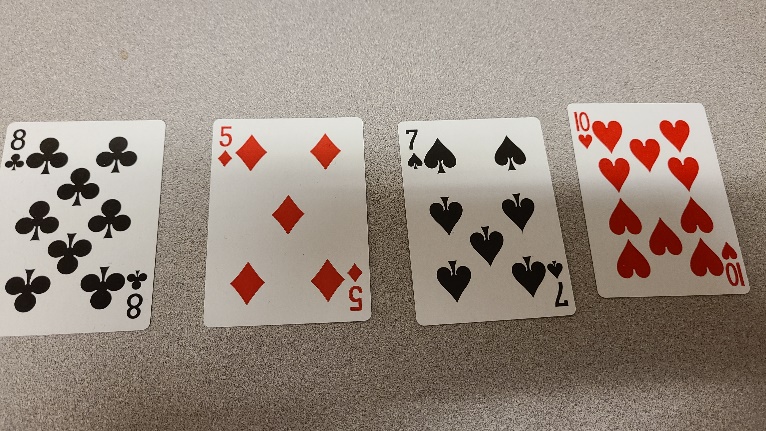


**unshuffled cards**

First, we shuffle the cards into a stack.



**Stack of shuffled cards**

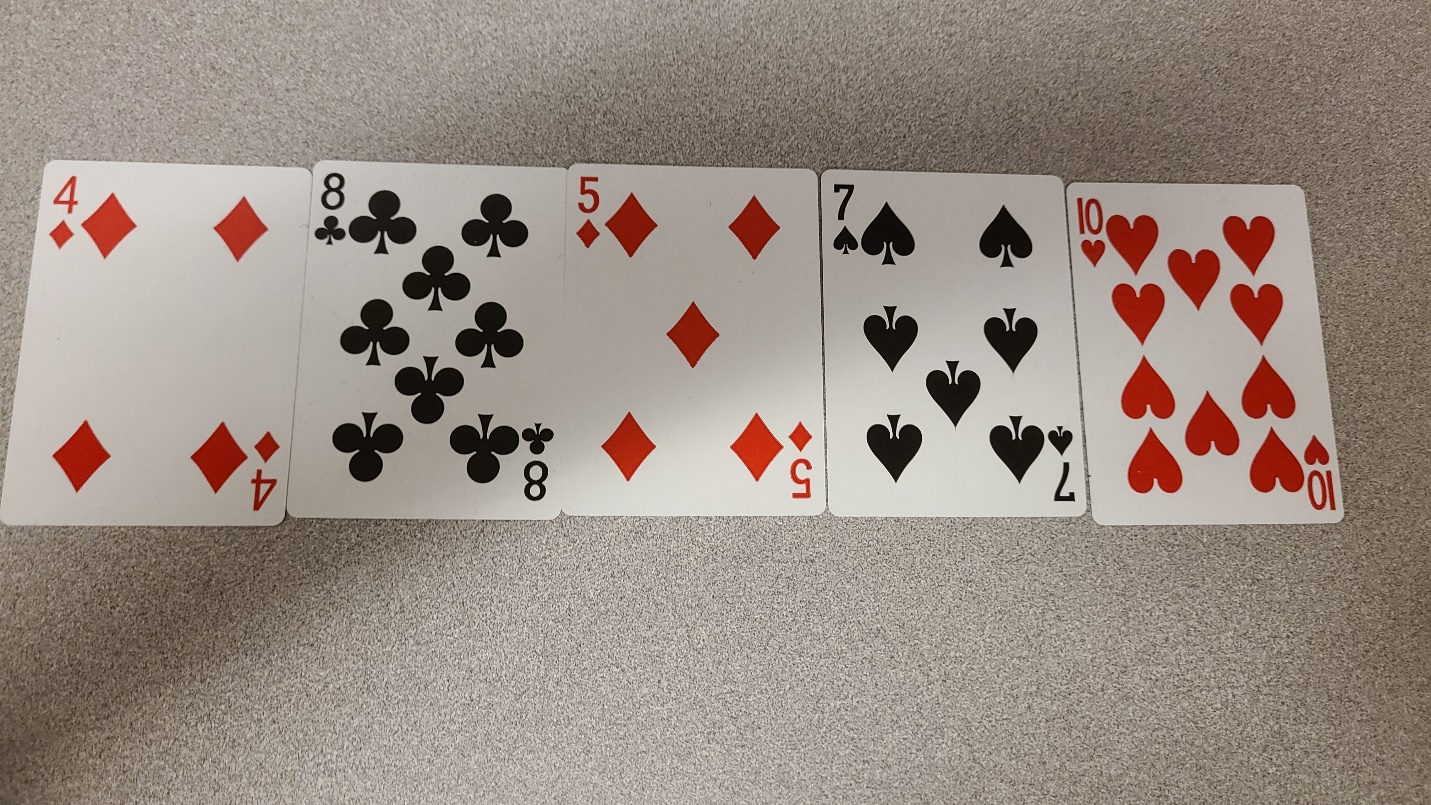
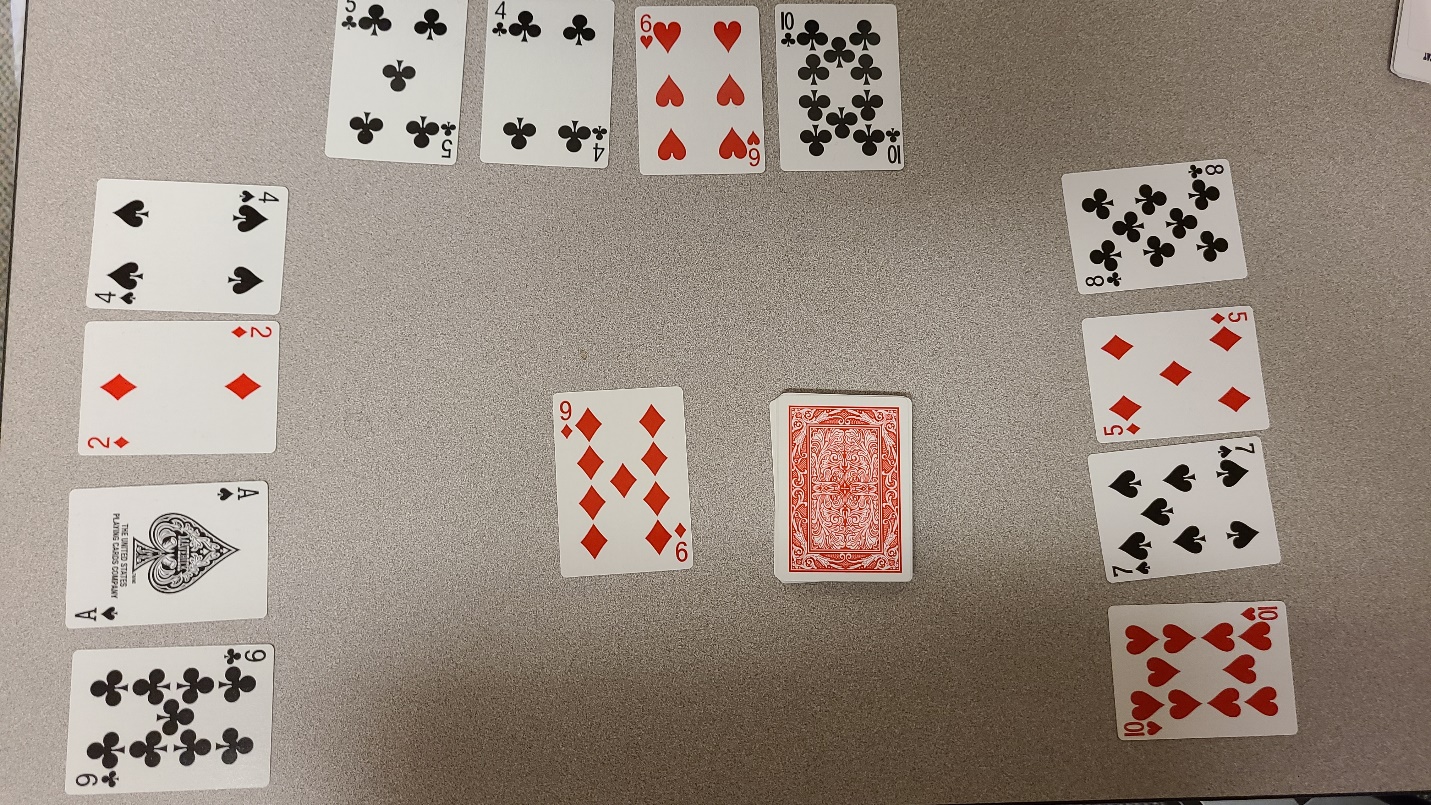
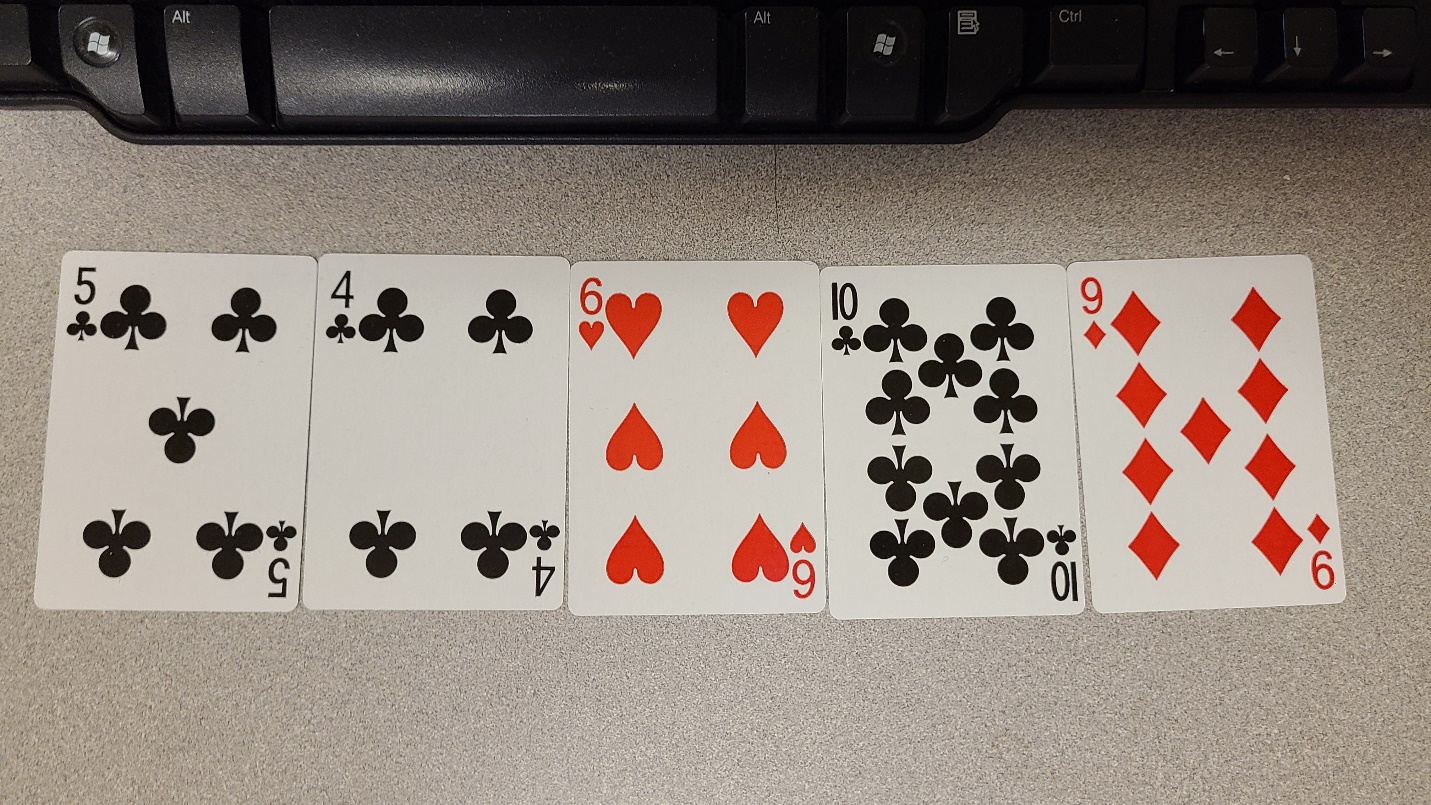
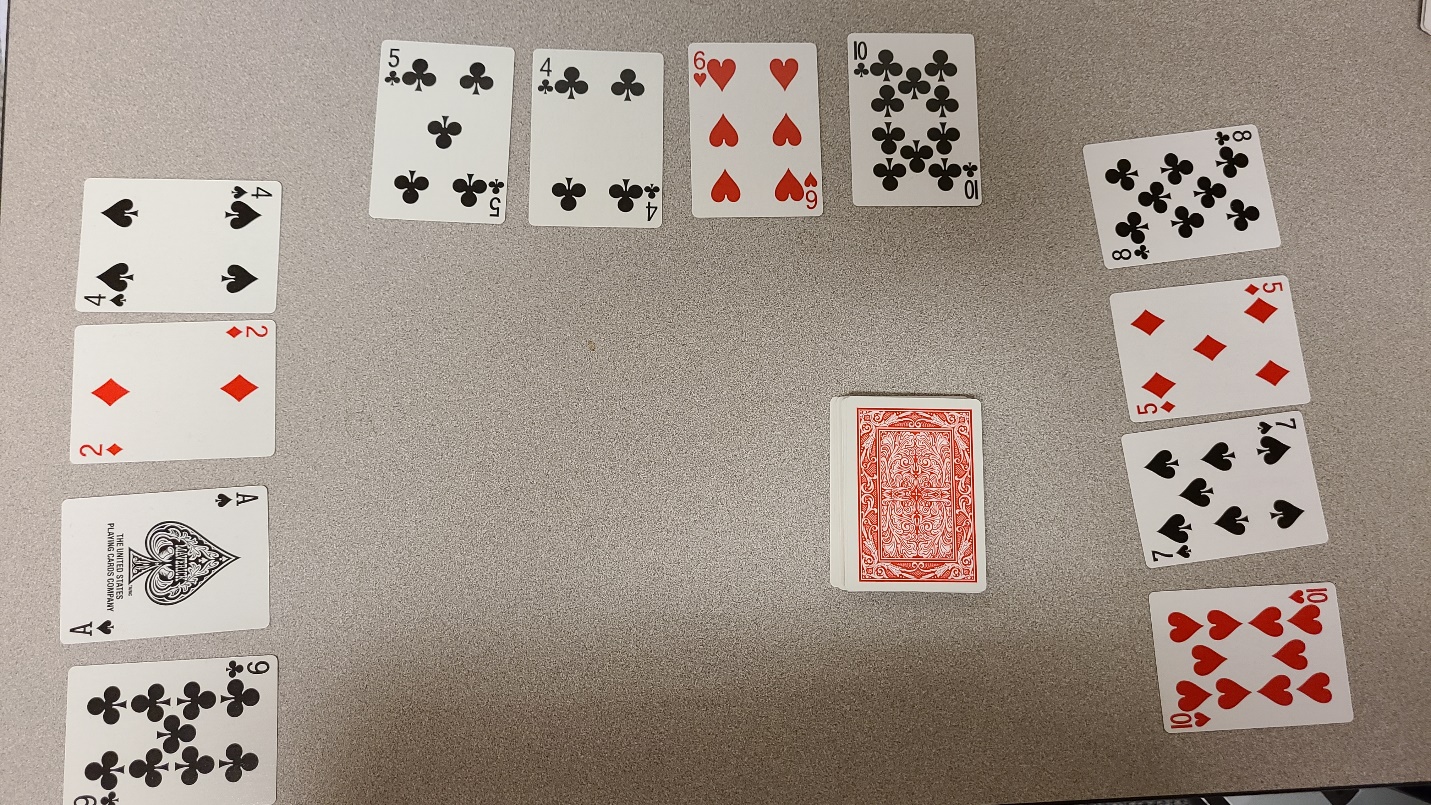


* 3 players are playing in this simulation.
* Each player will be dealt 4 cards.
* The cards are shown to help you understand the game (Players cannot see each other’s cards and cannot make decisions based on other players’ cards)

**Player2 cards**

**Player1 cards**

**Player3 cards**



* Player 1 starts by drawing a card form the faced down stack of cards
* If the 4 of the cards are of the same type or add up to 20 they will win
* Since that’s not the case, player 1 will throw the largest card that is not part of a series of same type cards
* The player has 3 club cards, 1 diamond and 1 hearts
* The 9 of diamonds is greater than the 6 of hearts
* The player will throw the 9 of diamonds

**Player1 cards after drawing from the stack**

**Stack of shuffled cards**

**Player3 cards**

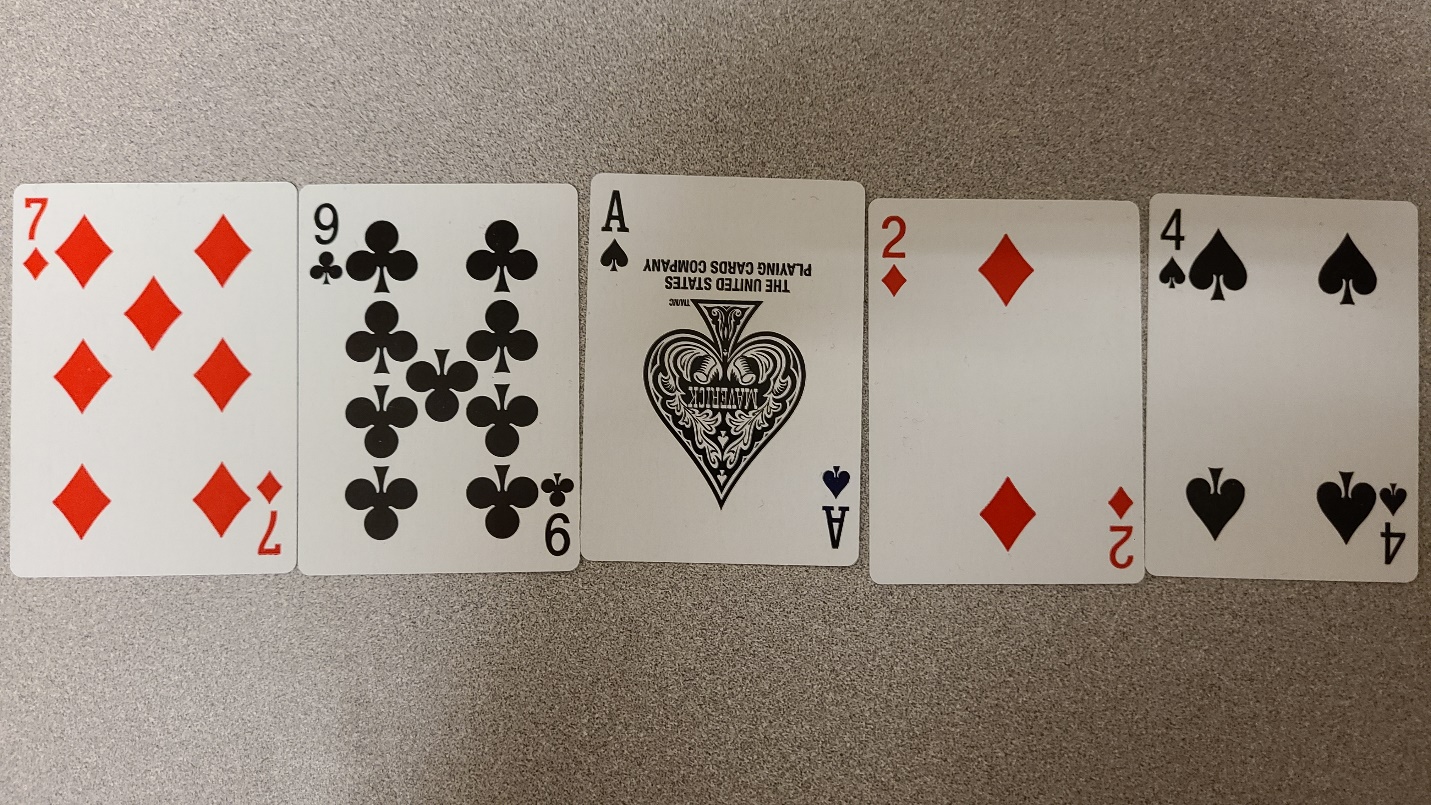
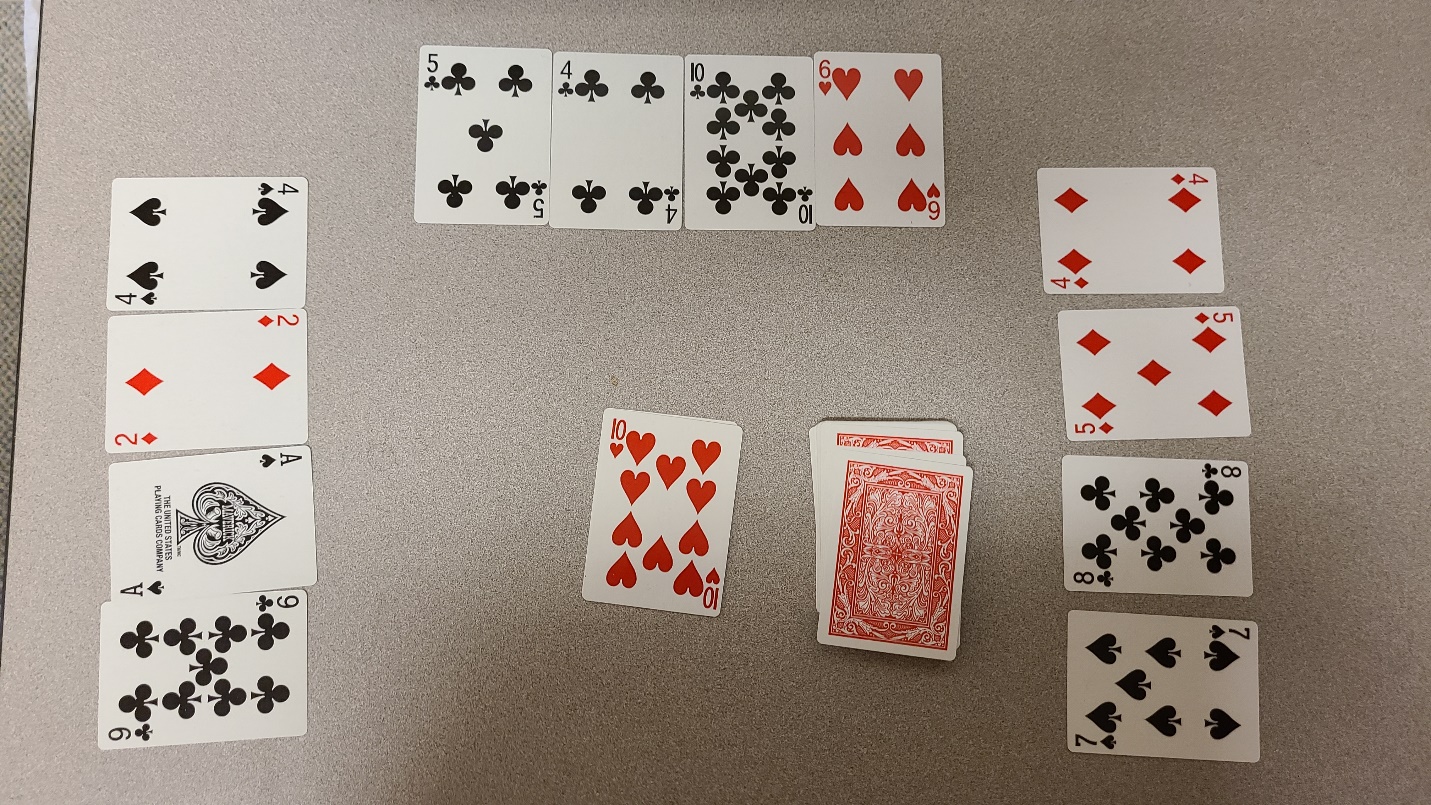
**Player2 cards**

**Player1 cards**

**Player1 throws 9 of diamonds to the floor**

**Player2 cards after drawing from the stack**

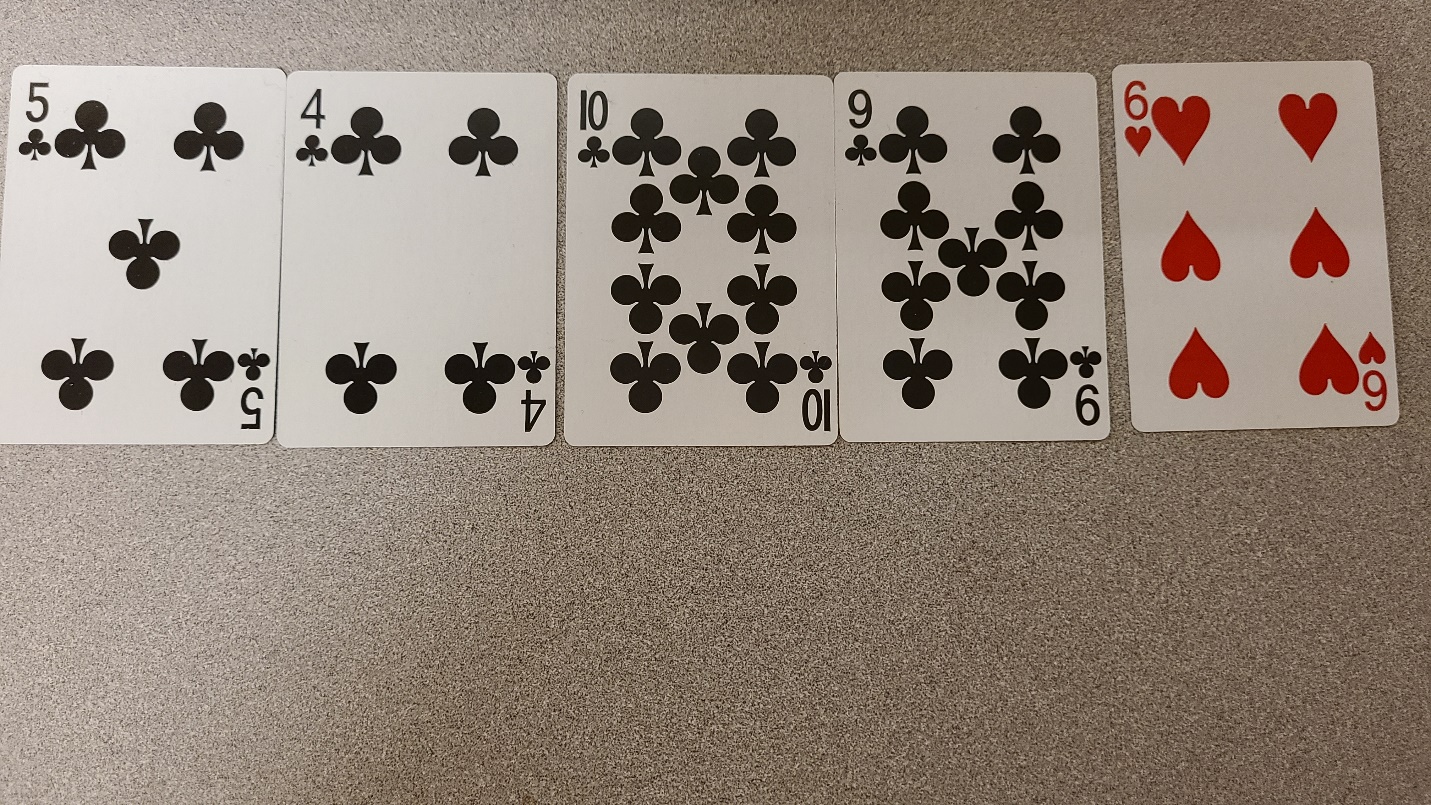
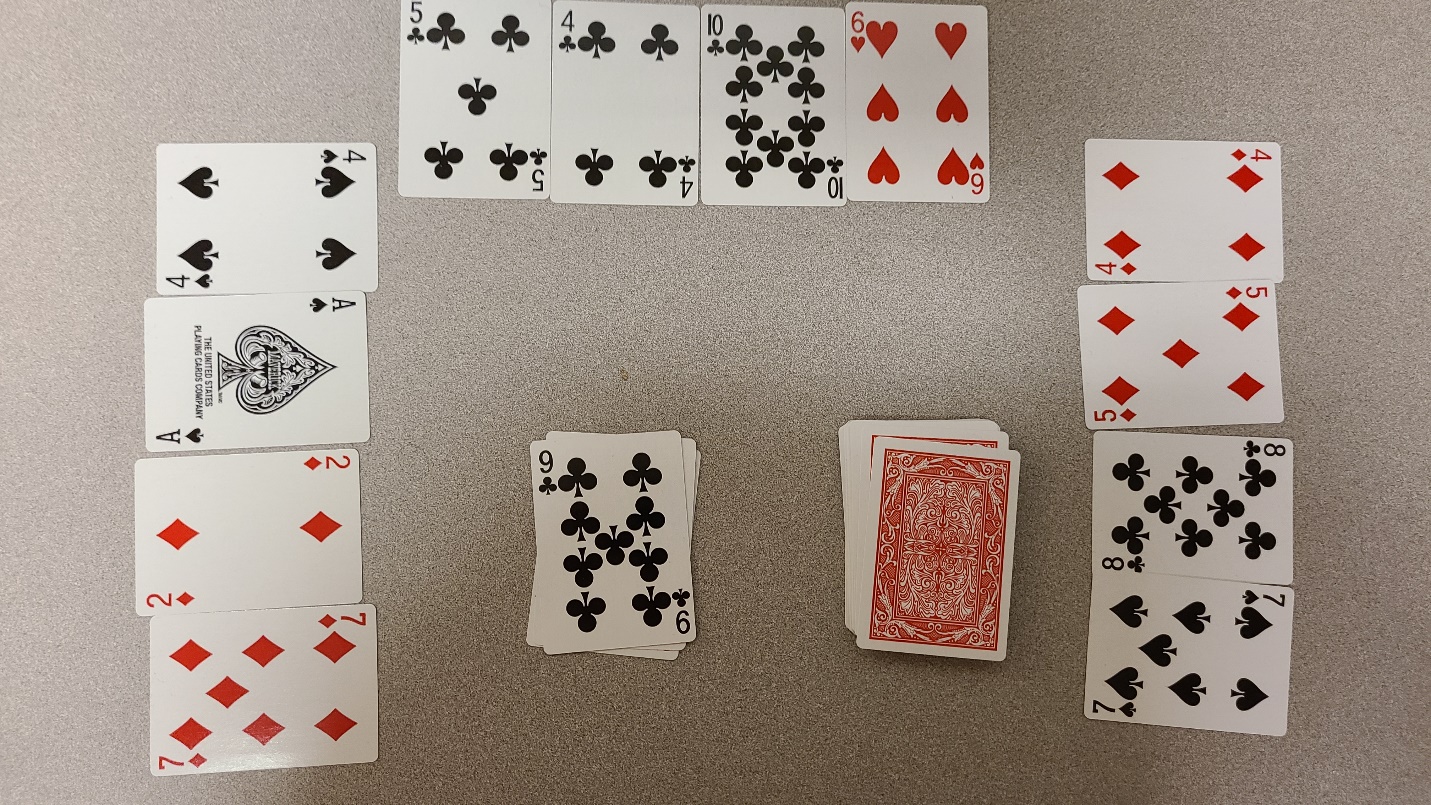
* Player 2 checks the card on the floor. The card on the floor will not make them win. So player 2 draws a card form the faced down stack of cards
* There are no 4 of cards that are of the same type or add up to 20
* Player 2 will throw the largest card that is not part of a series of same type cards
* The player has 2 diamond cards, 1 clubs, 1 spades and 1 hearts
* The 10 of hearts is largest.
* The player will throw the 10 of hearts



**Player3 cards after drawing from the stack**

**Player2 throws 10 of hearts to the floor**

* Player 3 checks the card on the floor. The card on the floor will not make them win. So player 3 draws a card form the faced down stack of cards
* There are no 4 of cards that are of the same type or add up to 20
* Player 3 will throw the largest card that is not part of a series of same type cards
* The player has 2 diamond cards, 2 spade and 1 clubs
* Since there is only one clubs card the player will throw the 9 of clubs



* Player 1 checks the card on the floor. The card on the floor will make them have 4 club cards. So, player 1 takes the 9 of clubs to win the game

**Player1 takes 9 of clubs from the floor to win the game**

**Player3 throws 9 of clubs to the floor**