Summary:

This project represents a simple card game between 2 players. The winner will be decided based on win-counts of a player. A deck of cards is being represented by using a queue whose int values represent a card. Values 0 to 12 represent cards from ace to king of suit ‘Club’. Values 13 to 25 represent cards from ace to king of suit ‘Diamond’. In the same manner, values from 26 to 38 and 39 to 51 represent cards from ace to king of suit ‘Heart’ and ‘Spades’ respectively. These values are saved as ranks. The rank of king is being treated as the highest. Similarly, the priority of the suits is as follows-

Club < Diamond < Heart < Spades

In our game, both players will draw a card and the winner of that set will be decided on the value of the cards.

Game Rules:

1. There will be two players in this game. Each player will get three cards from a well shuffled deck. Each player draws a card in each set and a winner is decided who is having a card of higher rank.
2. If the ranks of both the cards is same, then the suits of the cards are compared. The one having a higher priority suit wins the set. For example, if player1 draws an ace of club and player2 draws an ace of spades, then player will win since spades is having a higher priority than club.
3. Out of three sets whoever wins two sets wins the match.

The code consists of three files namely Prog1.java, IntStack.java, IntQueue.java. All the files are in a package called ‘hw03’. IntQueue.java contains the implementation of a stack data structure using java arrays. Similarly, IntQueue contains the implementation of a queue data structure using java arrays. Prog1.java is our main program which contains the main method.

The main method first calls the create_cards() method which assigns the values to an array. This array is then shuffled using method shuffle(). These cards are then enqueued into the ‘deck’ queue. This is then followed by calling redistribute() function which dequeues total six cards from the deck queue and pushes them onto player1 stack and player2 stack. Finally, method draw_cards() is called which implements the comparing logic for our game. The successful call of this function provides us with a winner. After completion of each set the cards are enqueued again to the deck queue.

In the first test, player1 gets cards 8, 9 and 3 which are eight of clubs, nine of clubs and three of clubs. Player2 gets 49, 25 and 5 which are 10 of spades, queen of diamond and five of clubs. In the first set, we compare three of clubs with five of clubs and the winner is five of clubs(Player2). In the second set nine of clubs is compared with queen of diamond and the winner is queen of diamond(Player2). Similarly, the third set is also won by player2. Hence, player2 wins the match.

In the second test, player1 gets 49, 28 and 44 which are ten of spades, two of hearts and five of spades. Player2 gets 36, 22 and 18 which are ten of heart, nine of diamond and five of diamond. In the first set five of spades is compared with five of diamond. Since the rank of both the cards is same we compare the suits. Spades is having a higher priority than diamond and hence five of spades wins the set(player1). In the second set two of heart is compared with nine of diamond and nine of diamond wins(player2). In the third set, ten of spades is compared with ten of hearts. Since both cards have the same rank we look for priority of suites. Since spades is having a higher priority than heart, ten of spades(player1) wins the set. Player1 wins this match by 2-1.