

```

import java.util.Random;
import java.util.Scanner;

public class BlackJack {
    //Gabriel Bostic and Uziel Ruelas Ramos (Clayton Romine and Garret Clark)

    private static final String[] SUITS = { "Hearts", "Diamonds", "Clubs", "Spades"
};
    private static final String[] RANKS = { "2", "3", "4", "5", "6", "7", "8", "9",
"10", "Jack", "Queen", "King",
    "Ace" };
    private static final int[] DECK = new int[52];
    private static int currentCardIndex = 0; //int outside of main
    //our ints
    private static int gamesWon = 0;
    private static int gamesLost = 0;
    private static int gamesTied = 0;


    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        //start loop here
        int i = 0;

        while (i == 0){
            //start of precode
            initializeDeck();
            shuffleDeck();

            int playerTotal = dealInitialPlayerCards();
            int dealerTotal = dealInitialDealerCards();

            playerTotal = playerTurn(scanner, playerTotal);
            if (playerTotal > 21) {
                System.out.println("You busted! Dealer wins.");
                gamesLost++;
            }

            else {
                dealerTotal = dealerTurn(dealerTotal);
                determineWinner(playerTotal, dealerTotal);
            }

            //end of precode
            System.out.println("Current Scores: ");
            System.out.println("Games Won: " + gamesWon);
            System.out.println("Games Lost: " + gamesLost);
            System.out.println("Games Drawn: " + gamesTied);
            System.out.println("Do you want to play again? Enter Y to play
again!");

            String playAgain = scanner.nextLine();
            if (playAgain.equals("Y")){
                i = 0;
            }
            else {
                i = 1;
            }
        }
    }
}

```

```

        //loop ends here
    }
    scanner.close();
}

private static void initializeDeck() {
    for (int i = 0; i < DECK.length; i++) {
        DECK[i] = i;
    }
}

private static void shuffleDeck() {
    Random random = new Random();
    for (int i = 0; i < DECK.length; i++) {
        int index = random.nextInt(DECK.length);
        int temp = DECK[i];
        DECK[i] = DECK[index];
        DECK[index] = temp;
    }
    System.out.println("printed deck");
    for (int i = 0; i < DECK.length; i++) {
        System.out.println(DECK[i] + " ");
    }
}

private static int dealInitialPlayerCards() {
    int card1 = dealCard();
    int card2 = dealCard();
    System.out.println("Your cards: " + RANKS[card1] + " of " +
SUITS[DECK[currentCardIndex] % 4] + " and "
        + RANKS[card2] + " of " + SUITS[card2 / 13]);
    return cardValue(card1) + cardValue(card2);
}

private static int dealInitialDealerCards() {
    int card1 = dealCard();
    System.out.println("Dealer's card: " + RANKS[card1] + " of " +
SUITS[DECK[currentCardIndex] % 4]);
    return cardValue(card1);
}

private static int playerTurn(Scanner scanner, int playerTotal) {
    while (true) {
        System.out.println("Your total is " + playerTotal + ". Do you want
to hit or stand?");
        String action = scanner.nextLine().toLowerCase();
        if (action.equals("hit")) {
            int newCard = dealCard();
            playerTotal += cardValue(newCard);
            System.out.println("new card index is " + newCard);
            System.out.println("You drew a " + RANKS[newCard] + " of " +
SUITS[DECK[currentCardIndex] % 4]);
            if (playerTotal > 21) {
                break;
            }
        } else if (action.equals("stand")) {
            break;
        }
    }
}

```

```

        } else {
            System.out.println("Invalid action. Please type 'hit' or
'stand'.");
        }
    }
    return playerTotal;
}

private static int dealerTurn(int dealerTotal) {
    while (dealerTotal < 17) {
        int newCard = dealCard();
        dealerTotal += cardValue(newCard);
    }
    System.out.println("Dealer's total is " + dealerTotal);
    return dealerTotal;
}

private static void determineWinner(int playerTotal, int dealerTotal) {
    // Score Tracking here
    if (dealerTotal > 21 || playerTotal > dealerTotal) {
        System.out.println("You win!");
        gamesWon++;
    } else if (dealerTotal == playerTotal) {
        System.out.println("It's a tie!");
        gamesTied++;
    } else {
        System.out.println("Dealer wins!");
        gamesLost++;
    }
}

private static int dealCard() {
    return DECK[currentCardIndex++] % 13;
}

private static int cardValue(int card) {
    return card < 9 ? card + 2 : 10;
}

int linearSearch(int[] numbers, int key) {
    int i = 0;
    for (i = 0; i < numbers.length; i++) {
        if (numbers[i] == key) {
            return i;
        }
    }
    return -1; // not found
}
}

```