```
#!/usr/bin/perl

#Complex perl program
# Program to play Blackjack
#author Allyson DeRensis
# date February 5, 2017

use strict;
use warnings;
use 5.010;

#To shuffle the deck
use List::Util qw(shuffle);
use english qw(-no_match_vars);


my $anotherGame = 'y';

my @deckOfCards = qw{A A A A 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6 7 7 7 7 8 8 8 8 9 9 9 9 J J J J Q Q Q Q K K K
K};

#shuffle the deck
my @cards = shuffle @deckOfCards;

my $bet = 0;
my $bankroll = 0;
my $hand = undef;

my @playerCards = {};
my @dealerCards = {};

my $total = 0;
my $playerCardTotal = 0;
my $dealerCardTotal = 0;

#just helpful variables
my $card = undef;
my $card2 = undef;

my $A = 0;

my $line = undef;
my $line2 = undef;

#my @test = qw{A A A A Q}; #test
# print "@test"; #test

my $answer = undef;

#print @cards; # a test
#print "\n";

# my @test = qw{A A A A Q}; #test
# print "@test"; #test

#print @cards; # a test
#print "\n";

sub bet{
#Find the bankroll and hand totals stored in files
        open (LOGFILE, "perlProgram\\bankroll.txt") or die "$! I couldn't get at bankroll.txt";
                $line = <LOGFILE>;
                print "bankroll: $line\n";
                close LOGFILE;
                $bankroll = $line;

        open (LOGHAND, "perlProgram\\hand.txt") or die "$! I couldn't get at hand.txt";
                $line2 = <LOGHAND>;
                print "The last hand was:# $line2\n";
                close LOGHAND;
                $hand = $line2;

        #Set the bet
        print "How much would you like to bet?  ";
        chomp($bet = <STDIN>);

        if ($bet eq "") {
                print "You are betting \$0 on this game of BlackJack. \n";
        } else {
                print "You are betting $bet on this game of BlackJack. \n";
        }
        &hand;
}

&bet;
&deal;


#deal 2 cards
sub hand{
        @playerCards=splice(@cards, 0, 2);
        @dealerCards=splice(@cards, 0, 2);

        #add to the number of hands played
        $hand = $hand + 1;
        print "This is hand # $hand \n";

        &revealCards;
}


#print "@dealerCards"; #test
sub revealCards{
        print "Your cards are:  @playerCards\n";

        print "The dealers cards are: one hidden card and @dealerCards[1 .. $#dealerCards]\n";
        &totals;
```

```
}

sub totals {
	$dealerCardTotal = &calculateTotal ( @dealerCards );
	$playerCardTotal = &calculateTotal ( @playerCards );
#	print "$dealerCardTotal\n"; #test
	print "Your total is $playerCardTotal\n";
	}

sub calculateTotal {
	$total = 0;
	$A = 0;
	foreach (@_){

		if (($_ ne 'A') && ($_ ne 'J') && ($_ ne 'Q') && ($_ ne 'K')) {
			$total = $total + $_;
		}

		#assigns a value to the face cards and adds it to the total
		elsif (($_  eq 'J') || ($_  eq  'Q') || ($_  eq  'K')) {
			$total = $total + 10;
		}

		#puts an ace aside for the moment
		elsif ($_ eq 'A') {
#			state $dealerCardTotal;
			$A = $A + 1;
		}

	#assigns a value of 11 or 1 to an ace depending on the total of the other cards,
	# and deals with the occurance of up to 4 aces - however unlikely
	if (($A == 1) && ($total <= 10)){
			$total = $total + 11;
		}
		elsif (($A == 1) && ($total > 10)) {
			$total = $total + 1;
		}



	if (($A == 2) && ($total <= 9)){
			$total = $total + 12;
		}
		elsif (($A >= 2) && ($total > 9)){
			$total = $total + 2;
		}



	if (($A == 3) && ($total > 8)){
			$total = $total + 3;
		}
		elsif (($A == 3) && ($total <= 8)){
			$total = $total + 13;
		}
```

```
    if (($A == 4) && ($total <= 7)){
            $total = $total + 14;
        }
        elsif (($A == 4) && ($total > 7)){
            $total = $total + 4;
        }
    }
    return $total;
}

#set up the subroutines to deal a card
sub dealCardsPlayer{
    $card = shift(@cards);
    push @playerCards, $card;
    & revealCards;
    & totals;
    & deal;
}

sub dealCardsDealer{
    $card = shift(@cards);
    push @dealerCards, $card;
    & revealCards;
    & totals;
    & deal;
}


sub cardQuery {

    print "Would you like another card? Please type 'y' or 'n' :\n ";
    chomp($answer = <STDIN>);

    #give the user another chance to answer
    if (($answer ne "n") && ($answer ne 'y')) {
        print "Would you like another card? Please type 'y' or 'n' :\n ";
        chomp($answer = <STDIN>);
    }

    if ($answer eq 'y') {
        print "You said y\n";
        &dealCardsPlayer;
    }

    elsif ($answer eq 'n') {
        print "You said n\n";
        while  ($dealerCardTotal <= 16) {
            &dealCardsDealer;
        }

    &results;
    }
}
```

```
#deal more cards as needed/ requested
sub deal {

        if ($dealerCardTotal <= 16) {
                &dealCardsDealer;
        }

        if ($playerCardTotal < 21){
                &cardQuery;
        }

        &results;
}

#compare totals
sub results {

        if (($dealerCardTotal > 21) && ($playerCardTotal <= 21)) {
                &winner;
        }

        if (($playerCardTotal > 21) && ($dealerCardTotal <= 21)) {
                &loser;
        }

        if (($playerCardTotal > 21) && ($dealerCardTotal > 21)) {
                print "No one wins. \n";
                &dealer;
                &game;
        }

        if (($playerCardTotal > $dealerCardTotal) && ($playerCardTotal <= 21)) {
                &winner;
        }

        if (($playerCardTotal < $dealerCardTotal) &&  ($dealerCardTotal <= 21)) {
                &loser;
        }
}

#find out the dealer's hidden card
sub dealer {
        print "The dealers cards are @dealerCards \n";
        print "The dealer's total is $dealerCardTotal \n";
}

sub player {
        print "Your cards are @playerCards \n";
        print "Your total is $playerCardTotal \n";
}

sub winner {
        &dealer;
        &player;
        print "You are the winner.\n";


        $bankroll = $bankroll + $bet;
        print "Your bankroll is now : $bankroll \n";

        &game;
}

sub loser {
        &dealer;
        &player;
        print "You have lost.\n";

        $bankroll = $bankroll - $bet;
        print "Your bankroll is now : $bankroll \n";

      &game;
}

sub game {
        open (LOGFILE, ">perlProgram\\bankroll.txt") or die "$! error trying to write";
        print LOGFILE $bankroll,"\n";
        close LOGFILE;

        open (LOGHAND, ">perlProgram\\hand.txt") or die "$! error trying to write";
        print LOGHAND $hand,"\n";
        close LOGHAND;


        print "Would you like to play again? Please type 'y' or 'n.'\n";
        chomp($anotherGame = <STDIN>);

        if ($anotherGame eq 'y') {
                print "You typed y\n";

                &bet;

        }elsif ($anotherGame eq "n"){
                print "You typed n\n";
                print "See you later alligator.";
                exit;
        }

}
```