```
#Dr. Burkman
#Mod 10 Homework

#declare and fill the four suits for the deck
$spades = @(
'Ace of Spades', 'King of Spades',
'Queen of Spades', 'Jack of Spades',
'10 of Spades', '9 of Spades',
'8 of Spades', '7 of Spades',
'6 of Spades', '5 of Spades',
'4 of Spades', '3 of Spades',
'2 of Spades')

$diamonds = @(
'Ace of Diamonds', 'King of Diamonds',
'Queen of Diamonds', 'Jack of Diamonds',
'10 of Diamonds', '9 of Diamonds',
'8 of Diamonds', '7 of Diamonds',
'6 of Diamonds', '5 of Diamonds',
'4 of Diamonds', '3 of Diamonds',
'2 of Diamonds')

$clubs = @(
'Ace of Clubs', 'King of Clubs',
'Queen of Clubs', 'Jack of Clubs',
'10 of Clubs', '9 of Clubs',
'8 of Clubs', '7 of Clubs',
'6 of Clubs', '5 of Clubs',
'4 of Clubs', '3 of Clubs',
'2 of Clubs')

$hearts = @(
'Ace of Hearts', 'King of Hearts',
'Queen of Hearts', 'Jack of Hearts',
'10 of Hearts', '9 of Hearts',
'8 of Hearts', '7 of Hearts',
'6 of Hearts', '5 of Hearts',
'4 of Hearts', '3 of Hearts',
'2 of Hearts')

#declare the play suits
$play_spades = @()
$play_diamonds = @()
$play_clubs = @()
$play_hearts = @()

function new_deck ()
{
    foreach ($i in $spades)
    {
        $Global:play_spades += $i
```

```
    }
        foreach ($i in $diamonds)
    {
        $Global:play_diamonds += $i
    }
        foreach ($i in $clubs)
    {
        $Global:play_clubs += $i
    }
        foreach ($i in $hearts)
    {
        $Global:play_hearts += $i
    }
    $Global:spades_gone=0
    $Global:diamonds_gone=0
    $Global:clubs_gone=0
    $Global:hearts_gone=0
}

function remove_card ($array, $card_to_remove)
{
    $temp_array = @()
    for ($i=0; $i -lt $card_to_remove; $i++)
    {
        $temp_array += $array[$i]
    }
    for ($i=$card_to_remove+1; $i -lt $array.Count; $i++)
    {
        $temp_array += $array[$i]
    }
    return $temp_array
}

function get_card ()
{
    clear
    #check for valid input
    $cards_requested = Read-Host "How many cards would you like to draw from
…this deck?"
    if ($cards_requested -notmatch "^[+]?[0-9]" -or $cards_requested -ne
…[int]$cards_requested)
    {
        clear
        Write-Host "Invalid Option.  Press Enter to return to the main
…menu:"
        Read-Host
        return
    }

    #see if there are enough cards in the deck to meet the request
    #write-host $play_spades.Count $play_diamonds.Count $play_clubs.count
```

```
$play_hearts.Count
    $cards_remaining = $Global:play_spades.Count + $Global:play_diamonds.Count + $Global:play_clubs.Count + $Global:play_hearts.Count
    if ($cards_requested/1 -gt $cards_remaining)
    {
        clear
        write-host "There are only $cards_remaining cards left in the deck but you have requested $cards_requested."
        write-host "`r`nPress the Enter key to return to the main menu: "
        Read-Host
        return
    }
    Write-Host "Your cards are:`r`n"

    #loop and get the number of requested cards
    while ($cards_requested -gt 0)
    {
        if ($spades_gone -eq 1 -and $diamonds_gone -eq 1 -and $clubs_gone -eq 1 -and $hearts_gone -eq 1)
        {
            Write-Host "All the cards have been drawn from this deck"
            break
        }

        #get a random suit
        $suit = Get-Random -Minimum 0 -Maximum 4
        if ($suit -eq 0)
        {
            $suit_count = $play_spades.Count
            if ($suit_count -eq 0)
            {
                $spades_gone = 1
                continue
            }
            else
            {
                $card = Get-Random -Minimum 0 -Maximum ($suit_count)
                if ($play_spades.count -eq 1){
                Write-Host $play_spades
                $Global:play_spades = @()
                $cards_requested = $cards_requested - 1
                continue
            }
            Write-Host $play_spades[$card]
            if ($play_spades.count -gt 1)
            {
                $Global:play_spades = remove_card -array $play_spades -card_to_remove $card}
                $cards_requested = $cards_requested - 1
            }
```

```
        }
        if ($suit -eq 1)
        {
            $suit_count = $play_diamonds.Count
            if ($suit_count -eq 0)
            {
                $diamonds_gone = 1
                continue
            }
            else
            {
                $card = Get-Random -Minimum 0 -Maximum ($suit_count)
                if ($play_diamonds.count -eq 1){
                Write-Host $play_diamonds
                $Global:play_diamonds = @()
                $cards_requested = $cards_requested - 1
                continue
            }
            Write-Host $play_diamonds[$card]
            if ($play_diamonds.count -gt 1)
            {
                $Global:play_diamonds = remove_card -array $play_diamonds
-card_to_remove $card}
                $cards_requested = $cards_requested - 1
            }
        }
        if ($suit -eq 2)
        {
            $suit_count = $play_clubs.Count
            if ($suit_count -eq 0)
            {
                $clubs_gone = 1
                continue
            }
            else
            {
                $card = Get-Random -Minimum 0 -Maximum ($suit_count)
                if ($play_clubs.count -eq 1){
                Write-Host $play_clubs
                $Global:play_clubs = @()
                $cards_requested = $cards_requested - 1
                continue
            }
            Write-Host $play_clubs[$card]
            if ($play_clubs.count -gt 1)
            {
                $Global:play_clubs = remove_card -array $play_clubs
-card_to_remove $card}
                $cards_requested = $cards_requested - 1
            }
        }
```

```
        if ($suit -eq 3)
        {
            $suit_count = $play_hearts.Count
            if ($suit_count -eq 0)
            {
                $hearts_gone = 1
                continue
            }
            else
            {
                $card = Get-Random -Minimum 0 -Maximum ($suit_count)
                if ($play_hearts.count -eq 1){
                Write-Host $play_hearts
                $Global:play_hearts = @()
                $cards_requested = $cards_requested - 1
                continue
            }
            Write-Host $play_hearts[$card]
            if ($play_hearts.count -gt 1)
            {
                $Global:play_hearts = remove_card -array $play_hearts
-card_to_remove $card}
                $cards_requested = $cards_requested - 1
            }
        }




    }


Read-Host


}

#get the new deck for the first time
new_deck


while ($true)
{
    clear
    Write-Host ("
Welcome to the card deck simulator.

Please select from the following options:

    1.  Draw a selected number of cards from the current deck
    2.  Get a new deck of cards
    3.  Exit
```

```
")

$user_menu_choice = read-host "Option#"

if ($user_menu_choice -eq 1)
{
    get_card
}
elseif ($user_menu_choice -eq 2)
{
    new_deck
}
elseif ($user_menu_choice -eq 3)
{
    clear
    break
}
else
{
    clear
    read-host "That is not a valid selection.  Press Enter to continue"
}


}
```