**Forms-PHP Lab**--Write code to allow a student to submit a credit card number in exchange for a good grade in a class. (Just a joke!)

*The HTML page buyagrade.html will have a form that submits to sucker.php. Then  sucker.php will output a web page to confirms that the student's money was taken. Use the HTML file buyagrade.html to get started. Create an HTML form that POSTs its submitted data to a PHP program on a server.*

1. Modify the HTML file by turning part of its contents into an HTML form.
2. Give name attributes to the form controls so they will be sent as query parameters; the names are up to you.
3. Some form controls (such as radio buttons) need value attributes.
4. Fill the Section drop-down list with choices 1st through 8th.
5. Test your form parameters by temporarily setting its action attribute to: params.php (copy this from the first day of forms.)

*Now write the PHP page to handle the submitted form data. Start from to sucker.php.*

1. Tell your buyagrade.html to POST its data to sucker.php.
2. Modify sucker.php to display the name, credit card number and type (Visa or MasterCard).
3. For now, the page doesn't need to save this information on the server.
4. Embed variables' values in HTML using PHP expression blocks such as <?= $foo ?>.

*Modify your sucker.php page to save the submitted data to a file suckers.txt.*

1. Use the following format:

Ryan; 1st;1234123412341234;visa

Kevin W;2nd;5963109385987345;mastercard

Kimberly Sims;3rd;7328904328904902;mastercard

Sandy Frees;4th;4444100020003000;visa

1. Also change your page's output to show the complete contents of this file to the user. Place the file contents into an HTML <pre> element to preserve whitespace.
2. You can read and write files using [file\_get\_contents](http://www.php.net/file_get_contents" \t "_blank) and [file\_put\_contents](http://www.php.net/file_put_contents" \t "_blank).
3. If you see an error about "Permission denied", change the permissions to read, write and execute. Use <http://www.macinstruct.com/node/415> for help.

*Basic Data Validation*

1. Update sucker.php to verify that all parameters exist and that the user did not leave any blank.
2. You can check whether a particular parameter has been passed using the PHP function isset.
3. You can check whether an element of $\_GET or $\_POST is blank by comparing it to the empty string, "".
4. You can also use the element itself as a Boolean test to see if it is "truthy" or "falsy".
5. If the user did not fill in every field correctly, show an error message.
6. Give a "Try again" link that directs back to the initial buyagrade.html page.

*Update your sucker.php file to further check the validity of the credit card number*

1. Check to make sure the credit card number is composed of exactly 16 characters.
2. Check to make sure that a Visa card number starts with a 4 and a MasterCard number starts with a 5.
3. If the credit card doesn't pass the above checks, show an error
4. A valid credit card number passes a digit-sum test known as the Luhn checksum algorithm.
   1. Luhn's algorithm states that if you sum the digits of the number in a certain way, the total sum must be a multiple of 10 for a valid number. Systems that accept credit cards perform a Luhn test before contacting the credit card company for final verification.
   2. The algorithm for summing the digits is the following: Consider each digit of the credit card to have a zero-based index: the first is at index 0, and the last is at index 15. Start from the rightmost digit (index 15) and process each digit one at a time. For digits at odd-numbered indexes (15, 13, etc.), simply add that digit to the cumulative sum. For digits at even-numbered indexes (14, 12, etc), double the digit's value, then if that doubled value is less than 10, add it to the sum. If the doubled number is 10 or greater, add each of its digits separately into the sum.
   3. Sample data: <http://www.paypalobjects.com/en_US/vhelp/paypalmanager_help/credit_card_numbers.htm>

sum = 0

for (each digit of CC number, starting from the last index):

if index is odd:

add digit to sum

else:

double the digit's value

if doubled value < 10:

add doubled value to sum

else:

split doubled value into its two digits

add first digit to sum

add second digit to sum