Grading Rubric –
Tutorial 13, Review

Description	Pts	Your Score
1. Use your editor to open the co_cart_txt.html, co_cart_txt.js, co_credit_txt.html, and co_credit_txt.js files from the html13 > review folder. Enter your name and the date in the comment section of each file, and save them as co_cart.html, co_cart.js, co_credit.html, and co_credit.js respectively.	**4**	
2. Go to the co_cart.html file in your editor. Link the page to the co_cart.js file, loading the file asynchronously. Study the contents of the file and the cart form. Take note of the field names and IDs of the files in the form.	**4**	
3. Within the <form> tag for the cart form, add attributes to open the co_credit.html file using the get method when the cart form is submitted. Save your changes to the file.	**5**	
4. Go to the co_cart.js file in your editor. Directly below the initial comment section, add an event listener for the window load event that does the following when the page is loaded: a. Runs the calcCart() function. b. Runs the calcCart() function when the field value is changed. (Hint: Apply an onchange event handler to the modelQty field in the cart form.) c. Uses a for loop that loops through every option in the group of shipping option buttons, adding an event handler to run the calcCart() function when each option button is clicked.	**10**	
5. Create the calcCart() function to calculate the cost of the customer's order using field values in the cart form. Within the calcCart() function, do the following: a. Create a variable named orderCost that is equal to the cost of the espresso machine stored in the modelCost field multiplied by the quantity of machines ordered as stored in the modelQty field. Display the value of the orderCost variable in the orderCost field, formatted as U.S. currency. (Hint: Use the formatUSCurrency() function.) b. Create a variable named shipCost equal to the value of the selected shipping option from the group of shipping option buttons multiplied by the quantity of machines ordered. Display the value of the shipCost variable in the shippingCost field, formatted with a thousands separator and to two decimals places. (Hint: Use the formatNumber() function.) c. In the subTotal field, display the sum of orderCost and shipCost formatted with a thousands separa- tor and to two decimal places. d. Create a variable named salesTax equal to 0.05 times the sum of the orderCost and shipCost vari- ables. Display the value of the salesTax variable in the salesTax field, formatted with a thousands separator and to two decimal places.	**16**	

e. In the cartTotal field, display the sum of the orderCost, shipCost, and salesTax variables. Format the value as U.S. currency. f. Store the label text of the shipping option selected by the user from the shipping field in the hidden shippingType field.		
6. Save your changes to the file and then open co_cart.html in your browser. Verify that the page correctly calculates and displays the total cost of the order as shown in Figure 13-63 and that the totals are automatically updated as you change the order options.	**6**	
7. Go to the co_credit.html file in your editor. Link the page to the co_credit.js file, loading the file asynchronously. Study the contents of the file and the forms and fields it contains. Close the file, saving your changes.	**6**	
8. Go to the co_credit.js file in your editor. Create an event listener for the window load event that retrieves the field values attached to the query string of the page's URL. Add the following to the event listener's anonymous function: a. Create the orderData variable that stores the query string text from the URL. Slice the orderData text string to remove the first ? character, replace every occurrence of the + character with a blank space, and decode the URI-encoded characters. b. Split the orderData variable at every occurrence of a & or = character and store the substrings in the orderFields array variable. c. Write the following values from the orderFields array into the indicated fields of the order form: i. orderFields[3] into the modelName field ii. orderFields[5] into the modelQty field iii. orderFields[7] into the orderCost field iv. orderFields[9] into the shippingType field v. orderFields[13] into the shippingCost field vi. orderFields[15] into the subTotal field vii. orderFields[17] into the salesTax field viii. orderFields[19] into the cartTotal field	**20**	
9. Add another event listener for the window load event that runs different validation event handlers when the page is loaded by the browser. Add code to the anonymous function for the load event that does the following: a. Runs the runSubmit() function when the subButton is clicked. b. Runs the validateName() function when a value is input into the cardHolder field. c. Runs the validateNumber() function when a value is input into the cardNumber field. d. Runs the validateDate() function when a value is input into the expDate field. e. Runs the validateCVC() function when a value is input into the cvc field.	**14**	
10. Create the runSubmit() function that is run when the form is submitted. Within the function, add commands to run the validateName(), validateCredit(), validateNumber(), validateDate(), and validateCVC() functions.	**4**	

11. Create the validateDate() function. The purpose of this function is to validate the credit card expiration date stored in the expDate field. Within the function, insert an if-else structure that tests the following: a. If no value has been entered for the expiration date, set the custom validation message to "Enter the expiration date". b. If the expiration date does not match the regular expression pattern: /"(0[1-9]|l[0-2])\/20[12]\d$/ set the custom validation message to "Enter a valid expiration date". (Hint: Use the test() method.) c. Otherwise set the custom validation message to an empty text string.	**9**	
12. The remaining functions have already been entered for you. Save your changes to the file.	**1**	
13. Return to the co_cart.html file in your browser and enter a sample customer order and click the checkout button to submit the order and load the co_credit.html file. Verify the following: a. The field values from the customer order are displayed in the order form. b. You cannot submit the payment unless you entered the name of the card holder, selected a credit card company, entered a valid credit card number (you can find lists of sample test credit card numbers on the web), entered a valid expiration date, and entered a valid CVC number.	**1**	
TOTAL	**100**	

YOUR SCORE: _____