



Program 4: JavaScript Error Checking

Student Learning Objectives

- Demonstrate the ability to author XHTML forms and tables
- Demonstrate the ability to use JavaScript error checking
- Demonstrate the ability to use PHP processing
- Demonstrate the ability to create valid, accessible, and semantically correct XHTML code

`program4.html`

Create a web page `program4.html` in folder `U:\htdocs\programs\program4\` which displays an XHTML form

- Displays a HTML form a user can fill out to order popcorn (see snapshot below)

- Uses JavaScript to perform error checking of all user input (validation rules below)
- Once all user input is correct, use the form's action attribute to redirect to `process.php` which calculates and displays the total amount owed for the popcorn sales

`program4.html`: default FireFox styles

ACM Club Popcorn Sales

Buyer's Name: * Required

Street Address: * Required

City, State, Zip: * Required

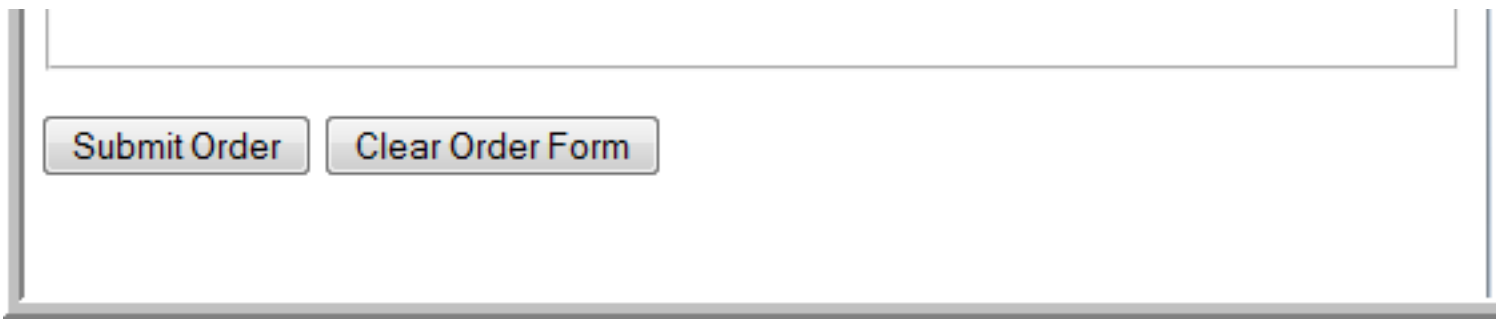
Products Available

Product Name	Price	Quantity
Unpopped Popcorn (1 lb.)	\$3.00	<input type="text"/>
Caramel Popcorn (2 lb. cannister)	\$3.50	<input type="text"/>
Caramel Nut Popcorn (2 lb. cannister)	\$4.50	<input type="text"/>
Toffee Nut Popcorn (2 lb. cannister)	\$5.00	<input type="text"/>

Payment Method

☐ Visa ☐ Master Card ☐ Check

Number:



program4.js JavaScript Validation Rules

Place your JavaScript validation rules in an externally linked file `program4.js`. Use simple alert boxes to display error messages. Return false if the user input is invalid; true if it validates correctly (true or false triggers whether the form's action attribute is executed)

Note: Write your own JavaScript error handling code; do not use jQuery or some other JavaScript library.

1. Buyer's name value conforms to the format of "Firstname Lastname"
 - The first name and last name must begin with uppercase letters
 - The first and last name must have at least one lowercase letter following the initial letter
 - There could be multiple spaces between the first name and last name
2. Street address value begins with a number, followed by one or more spaces, followed by one or two street names
 - Street names must begin with an uppercase letter, which must be followed by at least one lowercase letter
 - If the street name has two parts, they must be separated by at least one but not more than

two spaces

3. City, State, Zip

- City begins with an uppercase letter followed by at least one lowercase letter followed by an optional comma
- State begins with an uppercase letter followed by one upper or lowercase letter followed by an optional comma
- Zip must be exactly 5 digits

4. Quantities ordered are in the range of 0-99

- Quantity may be left blank, but at least one popcorn product must be ordered

5. Payment method is specified

- For Visa and MasterCard, the credit card number must have the form "dddd-dddd-dddd-dddd" where the dashes are optional
 - a valid credit card number either has all three dashes **or** no dashes
- For payments by check, no credit card number is entered

process.php

When the user input data validates completely, the form's action value should redirect to `process.php` that calculates and displays the total price owed.

- Display the total as currency

program4.html

ACM Club Popcorn Sales

Buyer's Name: * Required

Street Address: * Required

City, State, Zip: * Required

Products Available

Product Name	Price	Quantity
Unpopped Popcorn (1 lb.)	\$3.00	<input type="text"/>
Caramel Popcorn (2 lb. cannister)	\$3.50	2 <input type="text"/>
Caramel Nut Popcorn (2 lb. cannister)	\$4.50	<input type="text"/>
Toffey Nut Popcorn (2 lb. cannister)	\$5.00	3 <input type="text"/>

Payment Method

☒ Visa ☐ Master Card ☐ Check

Number:

Submit Order

Clear Order Form

process.php

ACM Club Popcorn Confirmation

Thank you for your order.

Total amount owed is \$22.00

Return to [program4.html](#)

Grading Criteria:

Note: Program assignments are **individual assignments** and should represent your own work. You should **not** use pair programming on the program assignments. If you received help with this assignment, acknowledge this help (who helped and the nature of the help) in your file's header comment.

Validate your generated HTML code (program4.html and process.php) as XHTML 1.0 Strict with no errors or warnings. They should also be checked for accessibility using the FAE Rule Set with no issues identified. Do **not** express stylistic information in the HTML page itself, such as inline styles or presentational HTML tags such as **b** or **font**.

Format your HTML, JavaScript, and PHP nicely so that it is readable following the [CS 250 coding standards](#). Place a comment header in each file containing your account, file, and

honor code. Include metadata on your HTML pages.

To Receive Credit

- A [scoring rubric](#) for program 4 will be used for grading purposes.
 - Turn in the [scoring rubric](#) [PDF] for grading purposes
 - Keep track of and include the **total completion time** (rounded to closest half-hour) it took to complete the program assignment and include the time on the rubric
 - Write a score **0 . . 4** in the rubric's **self assessment column** representing your completion status
 - Make sure your work is stored in your CS 250 account in folder
`U:\htdocs\programs\program4\`

Program 4 is due at the start of lab on Monday. **No** late programming assignments will be accepted without prior approval.