

## Lab 8 - Part 1: SHIPPING FORM CONFIRMATION

REQUIREMENTS: *Using the provided code process the shipping form.*

Note: This form will not redirect to a different page. For this exercise, we are using the same page to display the confirmation message, so your code must stop the form from submitting fully.

1. Upon submission of the form:
  - a. The form disappears
  - b. A thank you message appears (*see bottom of html page*).

## Lab 8 - Part 2: ADD THE USER DATA

REQUIREMENTS: *Continuing with the same file ...*

Now, display the input data from the user in the thank you message:

“Thank you **NAME** (Customer # **CUSTOMERID**) for your purchase.  
Your package will be sent to **POSTALCODE**.”

## Lab 8 - Part 3: VALIDATE THE FORM

REQUIREMENTS: *Continuing with the same file ...*

*Inside the form handling function...*

1. If the user does not provide any input for the name field, the background of the field turns red, the cursor is set there, and the form does not submit.
2. If the user’s input for the ID field or Postal Code field does not match the provided pattern, the background of the field turns red, the cursor is set there, and the form does not submit.
3. If no fields are invalid, the user sees the thank you message from Part 2.

### EXTRA CHALLENGE

Uncomment the *speed of delivery* form field and add validation for it.  
Your output message should now say:

**“Thank you *NAME* (Customer # *CUSTOMERID*) for your purchase.  
Your package will be sent by *LEVELOFDELIVERY* to *POSTALCODE*. Cost  
of shipping will be \$*COSTOFSHIPPING*.”**

Note: The cost of shipping for each level of service is provided as a  
`value` attribute in the `<option>` element.

### EXTRA *EXTRA* CHALLENGE

Following the strategies of object oriented programming, use the  
*shipInfo* object to store the values from the form, and then as the  
source for the output message.