

Develop the Income Tax Calculator

Create a web-based Income Tax Calculator that allows users to input their annual income and calculate their tax liability based on specified tax brackets.

Requirements:

1. HTML Structure:

- An input field for entering the annual income.
- A button to calculate the tax.
- A display area to show the calculated tax.
- Optional: Add fields for tax deductions or credits.

2. JavaScript Functionality:

- Use function expressions and arrow functions to handle calculations.
- Implement event listeners for button clicks.
- Use conditional statements to apply different tax rates based on income brackets.

3. CSS Styling:

- Apply basic styling to create a user-friendly interface. The design is flexible and can be modified as needed.

The screenshot shows a simple web-based income tax calculator. It has a blue header bar with the title "Income Tax Calculator". Below the header, there is a form with two input fields and a button. The first input field is labeled "Enter taxable income:" and contains the value "57650". To its right is a "Calculate" button. Below this, another input field is labeled "Income tax owed:" and contains the value "8473.00".

Tax Brackets (Example):

- 0% for income up to \$10,000
- 10% for income from \$10,001 to \$40,000
- 20% for income from \$40,001 to \$100,000
- 30% for income over \$100,000

Deliverables:

- A zipped folder containing:
 - HTML file
 - CSS file
 - JavaScript file

Bonus Challenges (Optional):

- Allow users to input deductions and adjust the taxable income.
- Add a reset button to clear the input and results.
- Implement local storage to save previous calculations.

Assignments Grading Rubric

	Unsatisfactory ($\leq 40\%$ of the points)	Satisfactory (60% of the points)	Good (80% of the points)	Excellent (100% of the points)
Requirements and Delivery (4 points)	<ul style="list-style-type: none">• Completed less than 70% of the requirements.• Delivered on time but not in correct format.	<ul style="list-style-type: none">• Completed between 70- 80% of the requirements.• Delivered on time, and in correct format.	<ul style="list-style-type: none">• Completed between 80-90% of the requirements.• Delivered on time, and in correct format.	<ul style="list-style-type: none">• Completed between 90-100% of the requirements.• Delivered on time, and in correct format.
Coding Standards (2 points)	<ul style="list-style-type: none">• No name, date, or assignment title included• Poor use of white space (indentation, blank lines).• Disorganized and messy• Poor use of variables (many ambiguous naming).	<ul style="list-style-type: none">• Includes name, assignment title. • White space makes program fairly easy to read.• Organized work.• Good use of variables (few unambiguous naming).	<ul style="list-style-type: none">• Includes name, and assignment title.• Good use of white space.• Organized work.• Good use of variables (no global variables, unambiguous naming)	<ul style="list-style-type: none">• Includes name, and assignment title.• Excellent use of white space.• Creatively organized work.• Excellent use of variables (no global variables, unambiguous naming).

Runtime (2 points)	<ul style="list-style-type: none"> • Does not execute due to errors. • User prompts are misleading or non-existent. • No testing has been completed, or no input validation. 	<ul style="list-style-type: none"> • Executes without errors. • User prompts contain little information, poor design. • Some testing or input validation has been completed. 	<ul style="list-style-type: none"> • Executes without errors. • User prompts are understandable, minimum use of symbols or spacing in output. • Most testing or input validation completed 	<ul style="list-style-type: none"> • Executes without errors excellent user prompts, good use of symbols, spacing in output. • Thorough and organized testing or input validation has been completed.
Documentation (2 points)	<ul style="list-style-type: none"> • Very limited or no documentation included. • Documentation does not help the reader understand the code. 	<ul style="list-style-type: none"> • Basic documentation has been completed including descriptions of all class variables. • Purpose is noted for each function 	<ul style="list-style-type: none"> • Clearly documented including descriptions of all class variables. • Specific purpose is noted for each function and control structure. 	<ul style="list-style-type: none"> • Clearly and effectively documented including descriptions of all class variables. • Specific purpose noted for each function, control structure, input requirements, and output results.