

# **CSC 440/540 Introduction to Computer Algorithms**

**Fall 2023**

## **Final Examination (Take-home)**

**Release Time:** Nov 21 6:00 PM

**Due Time:** Nov 30 11:59 PM

**Total points:** 150

**NAME** \_\_\_\_\_

**STUDENT ID#** \_\_\_\_\_

1. (50 points) **Short answer questions** (2 points for each question).

### **Part I: Web Programming Fundamentals**

- (1) Describe the relative advantages and disadvantages of web-based applications in comparison to traditional desktop applications.
- (2) What is the Internet Protocol (IP)? Why is it important for web developers?

### **Part II: HTML**

- (3) What role do HTML validators play in web development?
- (4) Describe the difference between a relative and an absolute reference. When should each be used?
- (5) What is the difference between HTTP GET and POST? What are the advantages and disadvantages of each?
- (6) What are the most common types of user input validation?

### **Part III: CSS**

- (7) What are the main benefits of using CSS?
- (8) What are the different parts of a CSS style rule?
- (9) What is the difference between a relative and an absolute measure unit in CSS? Why are relative units preferred over absolute units in CSS?
- (10) What are class selectors? What are id selectors? Briefly discuss why you would use one over the other.
- (11) Illustrate the CSS box model. Be sure to label each of the components of the box.
- (12) What is opacity? Provide examples of three different ways to set it in CSS.

- (13) Describe the differences between relative and absolute positioning.
- (14) Describe how block-level elements are different from inline elements.
- (15) In CSS, what does floating an element do? How do you float an element?

#### **Part IV: JavaScript**

- (16) What kind of variable typing is used in JavaScript? What benefits and dangers arise from this?
- (17) How are function declarations different from function expressions? Why are function expressions often the preferred programming approach in JavaScript?
- (18) What is callback function?
- (19) Identify and define three types of scope within JavaScript. Provide a short example that demonstrates these scope types.
- (20) What are some key DOM objects?
- (21) What are the five key DOM selection methods? Provide an example of each one.
- (22) Why is the event listener approach to event handling preferred over the other two approaches?

#### **Part V: React**

- (23) What is a single-page application? Why are frameworks helpful in their creation?
- (24) What is JSX? What benefits does it have for web developers?
- (25) How are functional components different from class components?

2. (25 points). **HTML** (5 points for each question).

\*\*\* **START PROJECT:** please find it in the **HTML** folder in this Final Exam package. \*\*\*

### PROJECT 3: HTML Site

DIFFICULTY LEVEL: Intermediate

#### Overview

This project is the first step in the creation of an art store website. Unlike the previous exercises, your task is to create an HTML page from scratch based on the image in Figure 3.34.

#### Instructions

1. Create **ch03-proj3.html**. The `<body>` should contain just seven `<img>` elements. The file **gallery-header.jpg** appears in the header of the page and then the six square images for each of the six galleries appear in the main section of the page.
2. Wrap each of the six square gallery images in a link to their respective page (e.g., **gallery1.png** to **gallery1.html**).
3. Create the six gallery pages. The content for each gallery page can be found in the **information.txt** file. Wrap the address information in an `<address>` element and make the link a working link to the correct page. Make the address and the highlights separate sections. The four highlight images for each gallery have the gallery name in the filename.
4. Make the image (**gallery-thin.png**) in the header of each gallery page a link back to the main **ch03-proj3.html** page.
5. In the information file, the latitude and longitude of each gallery is provided. These numbers can be used to accurately show the gallery on a map. Later in the book, you will learn how to do so directly via JavaScript. For now, you will simply add a link in the following format:  
**`https://maps.google.com/?q=LAT,LON`**  
where LAT and LON will be replaced with the latitude and longitude numbers from the information file.

#### Guidance and Testing

1. To remove spaces between smaller square museum images, put all the markup for those museum images and links on a single line. Remember that the browser interprets returns and tabs as white space.
2. Display **ch03-proj3.html** in a browser and test each of the links. Verify the map links work correctly.



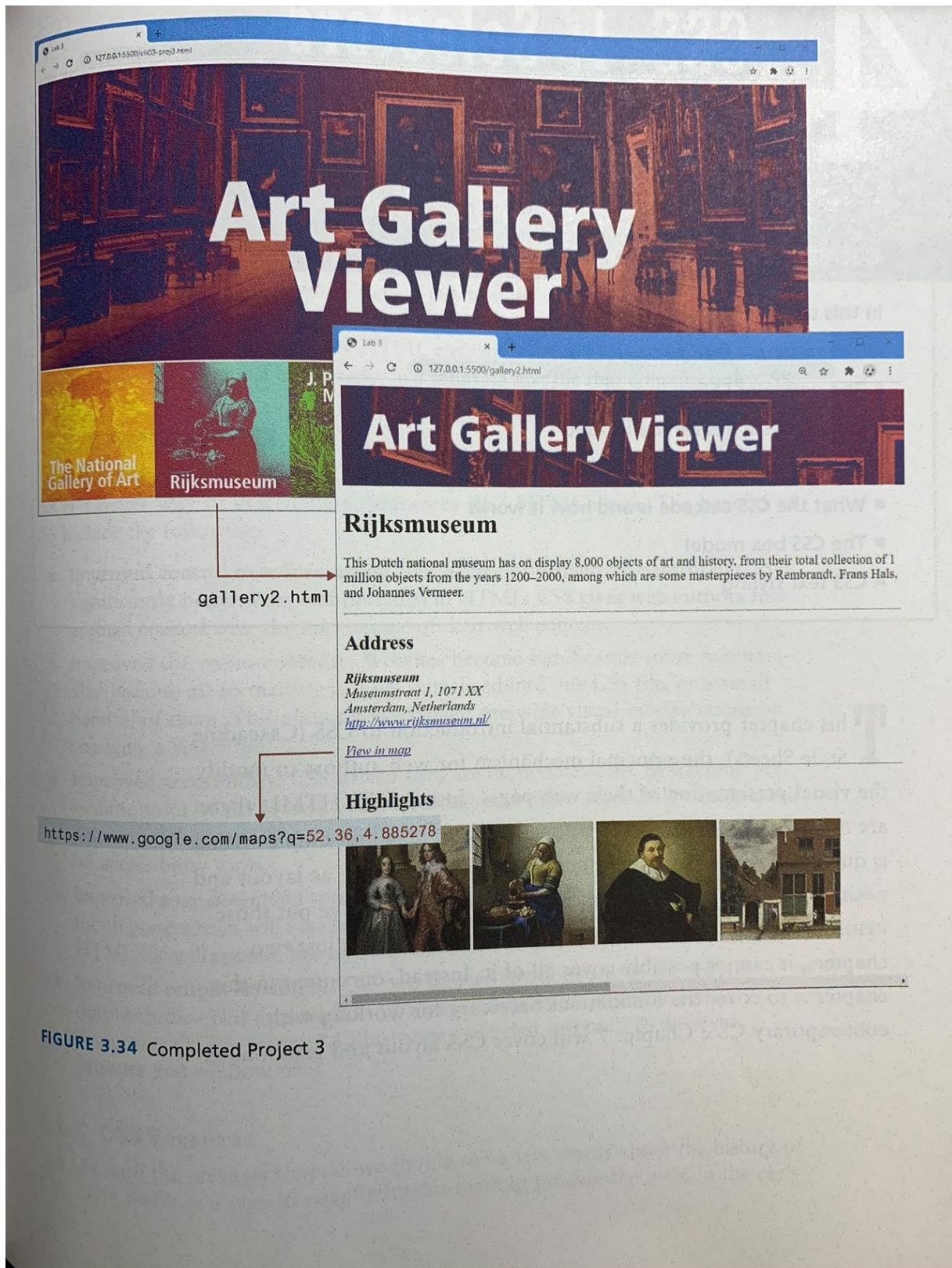


FIGURE 3.34 Completed Project 3

\*\*\* START PROJECT: please find it in the **HTML** folder in this Final Exam package. \*\*\*



3. (40 points). CSS (5 points for each question).

\*\*\* **START PROJECT:** please find it in the **CSS** folder in this Final Exam package. \*\*\*

**PROJECT 3: Home Page Prototype**

**DIFFICULTY LEVEL: Advanced**

**Overview**

In this project, you will make use of your knowledge of CSS to create a sample home page with navigation, large hero image, and three “card” boxes.

**Instructions**

1. Examine `ch04-proj03.html` in a browser and then in the editor of your choice. Do not make any changes to this file.
2. Edit the file `ch04-proj03.css` by defining styles so that it looks similar to that shown in Figure 4.43. The steps below provide more details.
3. The `<header>` will contain a background-image. Set its background-size to cover. Set the width of the `<header>` to 100% and its min-height to 500px.

background-image #51279B, #8662C7

font-sizes 48px, 24px

Use same colors as navigation

For each card, use inline-block

font-sizes 14, 12, 10px

background-image

**FIGURE 4.43** Completed Project 3

4. Add the logo in the top-left corner as a background-image to the `<nav>` element. Set its size to about 60px. The padding and height of the `<nav>` will also have to be set based on the size of the logo.
5. For each list item in the `<nav>` element, remove the list bullets by setting the `list-style-type` to `none`. Make the list horizontal by setting the `display` property of each `<li>` to `inline-block`. Set the link, visited, and hover colors of the navigation links.
6. Set the margin of the `<div>` within the `<header>` to position it roughly in the vertical middle of the big photo. Set its left margin so it is aligned with the navigation.
7. The card `<div>` elements need to be on a single line, so set the `display` property of each card to `inline-block`. For the `<div>` within the card (and its contents), set their padding and margins to get a similar appearance as Figure 4.43.
8. For the heart and comment `<span>` elements, use the background-image, background-size, padding, and margin properties to get a similar appearance as Figure 4.43.

#### Guidance and Testing

1. This project requires more styling changes, and so it is important to break it down into smaller steps. The instructions above help with this, but you could do many of these steps in a different order. Some developers like getting a small set of related elements styled correctly; others like to instead get the bigger structural elements styled first. You will have to find your own preferred approach.
2. Test each step along the way in a browser. It's not important that your page matches exactly the image shown in Figure 4.43. You are only trying to get it pretty similar.
3. Remember: many of our students struggle with CSS. It's normal if you struggle at times as well!

**\*\*\* START PROJECT: please find it in the CSS folder in this Final Exam package. \*\*\***



4. (35 points). **JavaScript** (5 points for each question).

**\*\* START PROJECT: please find it in the JavaScript folder in this Final Exam package.\*\***

### PROJECT 1: Art Store

DIFFICULTY LEVEL: Beginner

#### Overview

Demonstrate your proficiency with loops, conditionals, arrays, and functions in JavaScript. The final project will look similar to that shown in Figure 8.30.




#### Instructions

1. You have been provided with the HTML file (ch08-proj01.html) that includes the markup for the finished version. Preview the file in a browser.
2. Examine the data file `data.js`. It contains an array that we are going to use to programmatically generate the data rows (and replace the hard-coded markup supplied in the HTML file).
3. Open the JavaScript file `functions.js` and create a function called `calculateTotal()` that is passed a quantity and price and returns their product (i.e., multiply the two parameter values and return the result).
4. Within `functions.js`, create a function called `outputCartRow()` that has the following signature:  

```
function outputCartRow(item, total) {}
```
5. Implement the body of this function. It should use `document.write()` calls to display a row of the table using the passed data. Use the `toFixed()` method of the number variables to display two decimal places.

Replace markup with JavaScript loop using supplied array data

Replace markup with calls to functions

Product	#	Price	Amount
 Portrait of Martin Soolmans	3	\$75.00	\$225.00
 View of Houses in Delft	1	\$125.00	\$125.00
 Woman Reading a Letter	2	\$100.00	\$200.00
Subtotal			\$550.00
Tax			\$55.00
Shipping			\$0.00
Grand Total			\$605.00

Create function to output single cart row

Create functions to calculate these values

**FIGURE 8.30** Completed Project 1



Note: your browser may display a warning message in the console about avoiding `document.write`. You can ignore this for now (in the next chapter and lab, you will learn the correct way to add content using DOM methods).

6. Replace the three cart table rows in the original markup with a JavaScript loop that repeatedly calls this `outputCartRow()` function. Put this loop within the `ch08-proj01.js` file. Add the appropriate `<script>` tag to reference this `ch08-proj01.js` file within the `<tbody>` element.
7. Calculate the subtotal, tax, shipping, and grand total using JavaScript. Replace the hard-coded values in the markup with your JavaScript calculations. Notice that the tax and shipping threshold are input from the user, so that you can verify your calculations are working. The shipping amount should be \$40 unless the subtotal is above the shipping threshold, in which case it will be \$0.

#### Test

1. Test the page in the browser. Verify that the calculations work appropriately by changing the input values.

**\*\* START PROJECT: please find it in the JavaScript folder in this Final Exam package.\*\***

- **END**