CIS 111 WEB PROGRAMMING

111 Project 4

Michael Hennessy, Computer & Information Science, UO

CIS 111 Project 4. Due: 1700 Mon 7/14.

Programming JavaScript

The Problem description

Albert Einstein used to take greatdelight in baffling friends with this puzzle.

First, write the number 1089 on a piece of paper, fold it, and hand it to a friend for safekeeping.

What you wrote down is not to be read until you have completed your amazing mental feat.

Next, ask your friend to write down any three-digit number, emphasizing that the first and last digits must differ by at least two. Close your eyes or turn your back while this is being done.

After your friend has written down the three-digit number, ask your friend to reverse it, then subtract the smaller from the larger.

Example: 654 - 456 = 198.

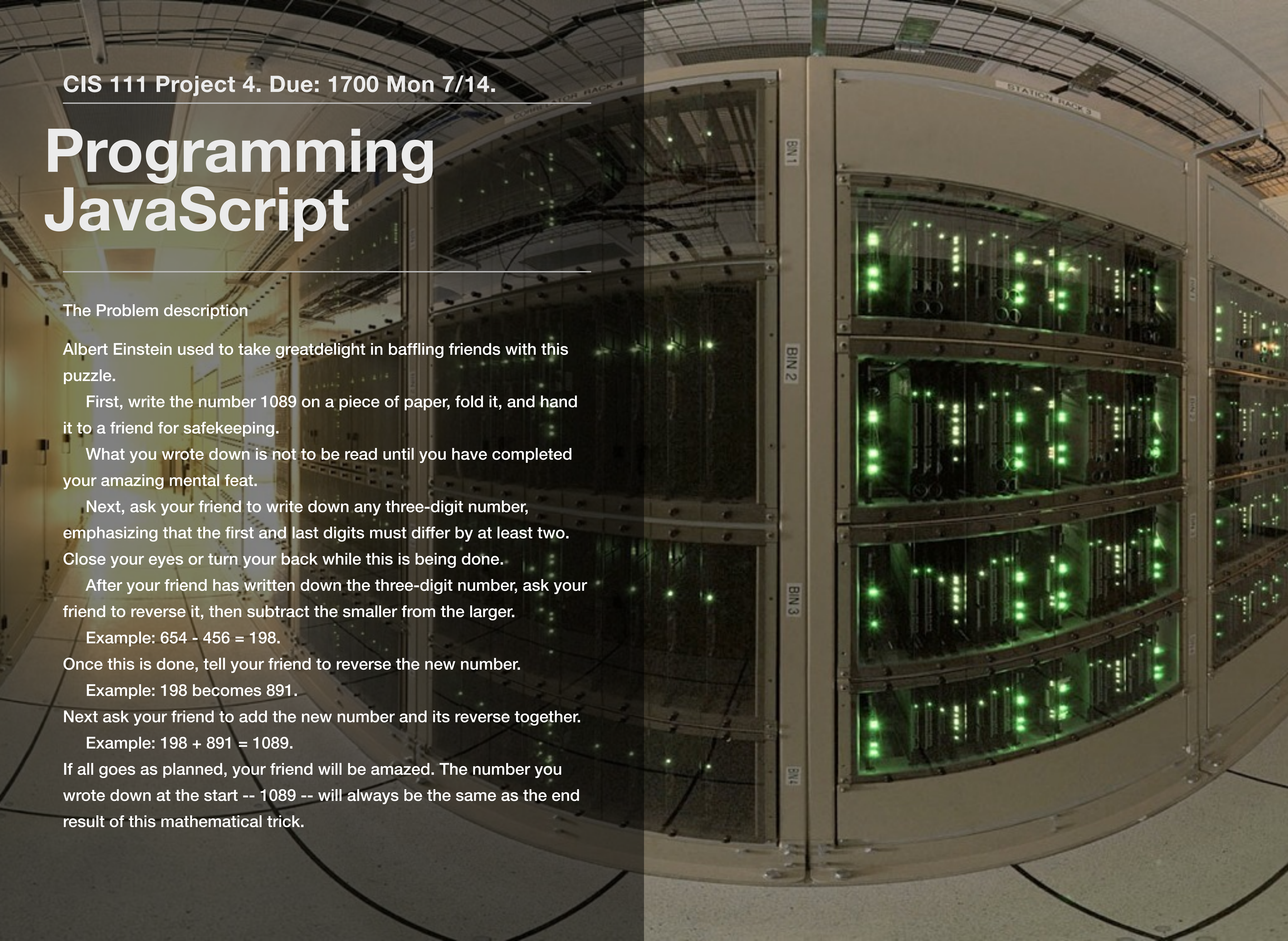
Once this is done, tell your friend to reverse the new number.

Example: 198 becomes 891.

Next ask your friend to add the new number and its reverse together.

Example: 198 + 891 = 1089.

If all goes as planned, your friend will be amazed. The number you wrote down at the start -- 1089 -- will always be the same as the end result of this mathematical trick.



Due: 1700 Mon 7/21.

PROJECT REQUIREMENTS

For the most efficient use of your time, solve the following problems in the order shown.

All the exercises on this project will be covered in class, with all your questions answered.

1. **[30 pts] 111/js/reverseN.js.**

   Read about the array join() method, the string split() method, and the array reverse() Method. These same methods are covered in ch. 3 of our textbook, JumpStart JavaScript.

   Note that Numbers do not have a reverse method, and Strings do not, but Arrays do. Therefore, if you convert a number to a String, then you can split it, reverse it, and join it to reverse the original number.

   parseInt and parseFloat both convert strings to numbers, but the simplest and most general way to do type conversions is by using the String() and Number() functions.

   Command-Line JavaScript: Use the above functions to write a function named reverseN that accepts a number n, converts it to a String, and returns a number that is the reverse of n.

   reverseN(123) => 321
   reverseN(1715) => 5171

   Use the DevTools JavaScript Console to test your function.

   Store the function definition in a file named reverseN.js in your 111/js/ folder.

2. **[30 pts] 111/p4/1089.html, and 111/js/1089.js.**

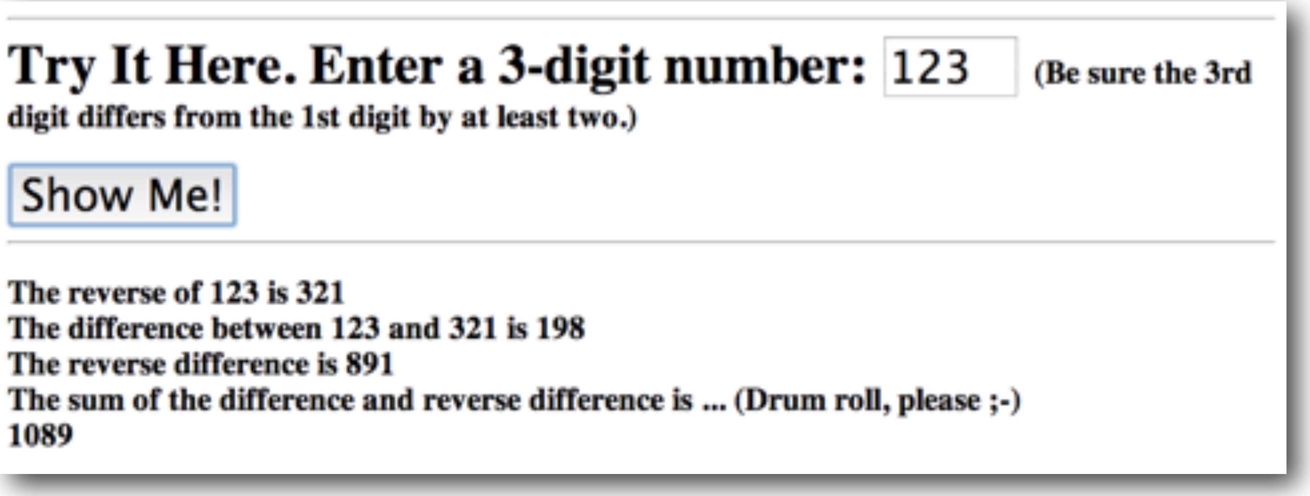
   Client-Side JavaScript: Download 1089.html from GitHub to your 111/p4/ folder.

   Create a new file named 1089.js in your 111/js/ folder.

1089.js should do the usual three steps:

a) Define an onclick handler named *driver*.
b) Grab the click button from 1089.html.
c) Register the onclick handler.

When the Show Me! button is clicked, the driver function generates five lines of output at the bottom of the web page:



Try It Here. Enter a 3-digit number: 123 **(Be sure the 3rd digit differs from the 1st digit by at least two.)**

The reverse of 123 is 321
The difference between 123 and 321 is 198
The reverse difference is 891
The sum of the difference and reverse difference is ... (Drum roll, please ;-)
1089

3. **[30 pts] 111/p4/deVerbalize.html, 111/js/deVerbalize.js, and deVerbalize.css.**

Problem Description. Your goal is to create a web app that converts alphbetic phone numbers into entirely numeric ones.

(A) Command-Line JavaScript. Write a function named charToNumber that accepts a character and returns the corresponding number on the keypad. Use the DevTools console to run the following (and other) tests:

charToNumber("A") => 2
charToNumber("z") => 9

(B). Command-Line JavaScript. Write a function named deVerbalize that converts alphbetic phone numbers into entirely numeric ones. deVerbalize accepts a string argument representing a telephone number (possibly containing letters) and returns a version of the phone number in which all letters have been replaced by their corresponding digits.

Use the DevTools console to run the following (and other) tests:

deVerbalize("1-800-IAm-Dave") => "1-800-426-3283"
deVerbalize("1-800-Comcast") => "1-800-2662278"

(C) Client-Side JavaScript. Download deVerbalize.html to your 111/p4/ folder. Download deVerbalize.css to your 111/css/ folder. Create a new file named deVerbalize.js in your 111/js/ folder. Write the JavaScript code to make the button work. All JavaScript goes in your .js file. No JavaScript goes in the .html file.

4. **[10 pts] 111/js/array-functions.js.**

   (A) Command-Line JavaScript. Write a function named allOdd that accepts an array of numbers and returns true if every number in the array is odd. Use the DevTools JavaScript Console to test the function. Store the function in 111/js/array-functions.js.

How to Turn In your Project

All you Have to Do is Make Sure your web pages are uploaded to the server by the Due-Date.

When your web pages are on the server, they can be graded.

You do not have to submit this project in Blackboard, nor do you have to notify your instructor in any way.

Just make sure you complete the project by the Due-Date, and do not upload or edit the files after the due-date. If you change the web page files in any way after the due-date, this will change the time-stamp of the files on the server, and your project will be late (zero points).

Project Grading Checkpoints

HOW YOUR PROJECTS WILL BE GRADED

- **There is no "submit your project in Blackboard" step for 111 projects.** When you have uploaded your web pages and tested them on the server, you are done turning in the project for grading.

- **The instructional staff has complete access to your project files for grading.** Your job is to make sure the files are on the server on time, and that you have tested them to make sure they are correct.

- **The files you upload to the server by the due-date are what will be graded**, so be sure to test your web pages on the server to make sure they are correct.

- **There are no second chances. Why? We do not have the time or the resources to grade your work twice.** Therefore make sure that what you upload to the server is correct. Test your web pages on the server after uploading them.

- **Time-Stamps are Crucial.** When you upload a file to the server, it is stamped with the exact time of the upload. This time-stamp must be no later than the project due-date. Your project is on-time only if the time-stamps show that it was uploaded to the server on time.

- **Do not re-upload any of your project files after the due-date.** If you do, this will change the time-stamp and your project will be late (0 pts).

- **Do not use Sublime's Sync feature, as this will change the time-stamp on all your files on the server.**

- **Your 111 folder on the server must be .htaccess password-protected.** If it is not, your project score will be zero (0). See your instructor or GTF for assistance if necessary.

- **Know the 111 Late Policy.**

Keeping these points in mind will help ensure you get full credit on your projects. Ask questions in class if anything is not clear.

Meeting the Deadline

How to Handle the 17:00 Deadline

- Start working on your project early.
- Friday Office and Help hours are jammed, and may end before you get assistance. Plan on completing all your projects before the deadline.
- Turn in what you have by the deadline-- partial credit is better than none.
- Piazza is good for answering verbal questions, but limited in terms of debugging help. For debugging, you need F2F help, which is the gold standard
- For Gold Standard Help:

  See *Contacts* in Blackboard.

  See *111 Help Hours* in Blackboard.

WebDev Workflow

Here is the CIS 111 Web Development Workflow:

1. **Edit**. Use the Sublime Text editor to create a web page (.html and .js files) on your computer.
2. **Preview**. Open the web page on your computer using Chrome. When it is perfect, and not before, go to the next step.
3. **Upload**. Move all project files (.html, .js, .png, etc.) to the server using an SFTP client (Sublime, CyberDuck, Aptana). This is also known as publishing the web page.
4. **Test**. Use Chrome to open your web page that is on the server. Do not use CyberDuck; do not use Aptana. Make sure that this web page is correct, because that is what will be graded.

Related Glossary Terms

Drag related terms here

Index Find Term