Programming

- Why is programming fun?
  - Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts and watching them work in subtle cycles, playing out the consequences of principles built in from the beginning. The programmed computer has all the fascination of the pinball machine or the jukebox mechanism, carried to the ultimate.


Schedule Changes

Labs 7 and 8:
- Both due next week
  - Tuesday, February 19, at 5pm
  - Rubric for Lab 8 will be available before lab tomorrow

Announcements

This week’s quiz:
- Chapters 20, 21, and 22 of Fluency
- Review
  - Questions at ends of chapters 20 and 21
  - Answers at back of book

This week’s quiz topics (continued)
- Variables—global and local
- Functions—syntax, names, declaring, calling, arguments, parameters,
- Loops—iteration variables, counters, step increase,
- Arrays—syntax, declaration, indexes, elements, using with the World-Famous Iteration
- Opening windows
- Dates
- Event handlers—onclick, onchange, onsubmit, onload, etc.
- Concatenation

Announcements

- This week’s quiz topics
  - Project and lab turn-ins
    - Catalyst Collect It shows date and time
    - Your html files show date and time
    - Do NOT keep working on your html files after the due date or they will be marked late!
Announcements

• Project turn-ins
  * 1-1-1 Rule (see Syllabus online)
    • One project part, such as Project 1A, can be one day late one time during the quarter
  • If you have used up your 1-1-1 rule for the quarter, turn in as much as you have finished so you get at least partial credit—rather than no credit!

• At end of quarter,
  * We will drop your
    • Lowest quiz score
    • Lowest lab score

Thinking Big: Programming Functions

A function is a package for an algorithm; once written, it can be used over and over.

Anatomy of a Function

• Functions are packages for algorithms
• 3 parts
  * Name
  * Parameters
  * Definition
• These parts are the function declaration

Pick a Name

• Name is the identifier for the function
  * Commonly used to describe what the function does
• Function declaration form:
  function <name>(<parameter list>)
  {
    <statement list>
  }

Parameters

• Parameters are the values the function will compute on, the input values
• They are given names
• Listed parameters are separated by commas
• Parameter names follow usual rules for identifiers

  function convertC2F (tempInC)
  {
    <statement list>
  }
Definition

- Definition is the algorithm written in a programming language
- To say what the answer/result is, JavaScript uses the statement: `return <expression>

```javascript
function convertC2F ( tempInC )
    return 9.0 / 5.0 * tempInC + 32;
}
```

- "Calling" a function is to run or execute it
  - Write the function's name, put the input values (arguments) in the parentheses:
    `convertC2F(38)`

Declaration versus Call

- A function's declaration is different from its call (use)
- Functions are declared once
- Functions can be called as many times as their answers are needed

Forms and Functions

- Construct a web page in which to run a function
- Recall `<form>` and `<input />` tags and event handlers in HTML
  - Event handlers usually implemented as functions
  - Using an input window, the value in that window can be used as an argument to a function

Calling to Customize a Page

- Three ways to get the result of a function call to print on the monitor
  1) Before the page is created
     - For example, with the `alert()` call (Fig. 20.1)
  2) Interactively after the page is displayed
     - For example, the Conversion application (Fig. 20.2)
  3) While the page is being loaded
     - For example, `document.write()` built-in function
- Calling functions while the browser is creating the page allows us to customize pages on-the-fly
Calling to Customize a Page

- How a browser builds a page:
  - Reads through HTML file, figuring out all tags and preparing to build page
  - Removes JavaScript tags and all text between them, and does whatever the JavaScript tells it to do
  - It could tell the browser to put some text back in the file, as in `document.write()`

- Suppose we want a table of temperature conversions for a web page with a column for Celsius and a column for Fahrenheit
  - Put `document.write()` within the `<script>` tags to create the rows of the table
  - Put Celsius values in first column cells, second column cells can call conversion function

Writing Functions, Using Functions

- Flipping Electronic Coins
  - A coin flip is an unpredictable event whose two outcomes are "equally probable"
  - Computers can generate pseudo-random numbers
    - An algorithm that produces a sequence of numbers that passes the statistical tests for randomness
    - We can just call them random numbers
  - Math.random() is JavaScript’s built-in function for generating random numbers
    - Each time it is called, it generates a random number between 0 (inclusive) and 1 (exclusive)
  - A function to flip electronic coins:
    ```javascript
    function coinFlip() {
      return Math.round( Math.random() );
    }
    ```
Flipping Electronic Coins (cont’d)

• coinFlip() returns with equal probability a 0 or a 1
• Next improvement is to return the text Heads or Tails rather than numbers

```javascript
function flipText() {
  if ( coinFlip() == 0 )
    return 'Tails';
  else
    return 'Heads';
}
```

Even more useful to give outcome in response to pressing a button on a web page

The Body Mass Index Computation

• BMI is a standard measure of weight in proportion to height
• Formula (in metric units):
  * Index = weight/height²
  Two parameters for this function, weight and height

```javascript
function bmiM(weightKg, heightCm) { // Compute BMI in metric
  var heightM = heightCm/100;       // Change cm to meters
  return weightKg / (heightM * heightM);
}
```

• Formula (in English units):
  * Index = 4.89 weight / height²
  Function:

```javascript
function bmiE(weightLbs, heightIn) { // Compute BMI in English
  var heightFt = heightIn / 12;      // Change inches to feet
  return 4.89 * weightLbs / (heightFt * heightFt);
}
```

• Function that could calculate BMI in type
  of units specified by user would need 3
  inputs (kind of unit, weight, height)

```javascript
function BMI(units, weight, height) {
  if (units == 'E')
    return bmiE(weight, height); // Answer in English
  else
    return bmiM(weight, height); // Answer in Metric
}
```

To put this function in a web page, we add radio buttons to select type
of units

• Two new features of radio buttons:
  * All related buttons share same name
    (clicking one on turns the other off)
  * Can be preset using checked='true'
• Add event handlers for the radio buttons
Scoping:

When to Use Names

• Scope of a name defines how “far” from its declarations it can be used

• General rule for scoping:
  • Variable names declared in a function can be used only within that function (they are local to the function)
  • Parameters are considered local variables
  • Variable names declared outside any function can be used throughout the program (global to the function)

Global/Local Scope Interaction

• Local variables come into existence when a function begins, and when it ends, they vanish
• Global variables are around all the time
• If information must be saved from one function call to the next, it must be in a global variable
Global/Local Scope Interaction (cont'd)

- y is globally declared and can be referenced anywhere
- y is also declared as a local variable in the tricky() function
- They are two different variables
- Which y is assigned the parameter x?
  - The locally declared variable, making it the "closest" declaration and hiding the globally declared y

The Memory Bank Web Page

- Create a web page for remembering useful computations and storing them in an interactive form
- Practice programming with functions

Plan the Memory Bank Web Page

- Each table row presents a computation
- Each text box except the last is an input to the computation
- The last text box is for the output
- Start with the row from the BMI computation page

Random Additions

- Add the row from the coin-flipping page
- Program event handler to keep track of the number of heads and tails flipped
- Use global variables so they keep their values across function calls
Revising Random Choice Function

• Write a function that chooses random whole numbers in a range from 0 to n, not including n
  
  function randNum( range ) {
    return Math.floor( range * Math.random() );
  }

• For coin-flipping, the range will be 2: 0 and 1
  
  randNum( 2 )

The Coin-Flipping Row

• Flip button and textboxes for current flip outcome, Heads total, and Tails total

• Use global variables to keep track of the number of heads and tails flipped
  
  • Increment appropriate variable with each flip

• Update/display current flip outcome and total number of heads or total number of tails with each flip

The "I’m Thinking of a Number" Row

• Guessing game – choose a number from 1 to n

• Use randNum() function, but shift the range by 1
  
  • randNum(n)+1;

• This table row is similar to coin-flipping row, but has a text box to set the upper end of the range
  
  • Declare global variable (topEnd) to say what the limit of the range is
  
  • When the user clicks button, the randNum() function is called with topEnd as the argument, and the result is incremented to shift its range. The result is displayed.

Improving the Memory Bank Web Page

• Needs to be fancier and include more features

• Program the memory bank to splash new pages onto the screen
  
  • Unlike a link, this allows both pages to display at the same time

Figure 20.9. The revised Memory Bank page and the Conversion page that displays when Convert is clicked.
A Counting Page

- To keep track of counts of things
- Counter Assistant application:
  - **Count** button increments **Total** field
  - **Meaning** field can be filled in with any text to remind us what the counter is
  - **C** clears all the fields for that row

Recap: Two Reasons to Write Functions

- Packaging algorithms into functions
- **Reuse**
  - Building blocks of future programming
  - Make them as general as possible
- Complexity management
  - Help us keep our sanity while we're solving problems

Add Final Touches to Memory Bank

- Add a date
  - JavaScript Date().toString()
  - References the date object, which contains the current date and time in numeric form, and converts to a printable form
- Add web links
  - Add any useful links (online dictionary, etc.) in their own column or in a row at the bottom of the table

Assess the Web Page Design

- Table data which spans two columns using colspan=2 attribute in <td> tag
- Links are grouped by topic
- Red bullet is used to separate entries
- Link area has a neat structure; adding new links is easy
FIT100 FIT100 FIT100
20-55

Figure 20.12 WTMD, for the link area of the Memory Bank Web page.

FIT100 FIT100 FIT100
20-56

Figure 20.13 Final version of the Memory Bank, Conversion, and Counter Assistant pages.