CMSI 585

PROGRAMMING LANGUAGES
(GRADUATE LEVEL)
Fall 2005

Assignment 1115

This assignment provides some firsthand experience with types, with a somewhat ironic twist — we're using JavaScript, where type rules are much looser and more dynamic than what you may be used to. But JavaScript *does* maintain types and track objects, and writing this program should give you some insight on how it does this.

Not for Submission

1. Read Scott Chapter 7.
2. Install and learn the JSUnit unit testing framework for JavaScript.

For Submission

As usual, submit code on hardcopy and by e-mail, and submit all other exercise responses on hardcopy. You are asked to write a "Thing Associator" in JavaScript, with an accompanying JSUnit test page (to be executed via JSUnit's *testRunner.html* page). The Thing Associator uses JavaScript's property-based approach to associate one "thing" with another. For example, you might associate "five" with 5.0, or 25 with "quarter." There are no restrictions on what gets associated what — you should be able to associate anything that JavaScript permits as a property key with anything that it permits as a property value.

1. Place the JavaScript code in a file called *thingAssociator.js*. Place your tests in a file called *yourname_thingAssociatorTest.html*. You may need to read up a little bit on JSUnit to figure out how it works; assorted examples have also been given in prior sample code.
2. Implement the following functions in *thingAssociator.js*:
   a. *function associate(map, key, value)*: This takes the given association object, *map*, and sets its *key* property to *value*. It shouldn't do anything fancy — it should set the property in the same way that standard JavaScript inline code would do it. We only define it as a common point of entry.
   b. *function getDirectAssociation(map, key)*: This takes the given association object, *map*, and returns the value that is currently associated with *key*. It should return the JavaScript *undefined* constant if *key* has not been associated with any value.
   c. *function getAssociationClosure(map, key)*: This takes the given association object, *map*, and recursively follows the *key* property within *map* until it reaches a value that is not itself a key in the map. This is the value that the function should return.
   d. *function reportAssociations(map)*: This "dumps" the *map* object's properties into a JavaScript array of "mapping" objects, numerically indexed starting at zero. Each mapping object should have four properties: *key*, which holds an association's property; *keyType*, which holds the type of that property; *value*, which holds the value associated with that key in the *map* object; and *valueType*, which holds the value's type.
3. Finally, test your Thing Associator through a JSUnit test page. As usual, precise interpretation of the specifications is critical, so don't hesitate to ask me for any clarifications.