

Review of Programming Concepts

1 Concepts

Here is a summary of the main components of programs that have been introduced this semester.

- **Variables.** Variables store data in a program. A variable has a name (also called an identifier). The types of data that can be stored in variables include numbers, strings (text), and arrays.

Typically, a *declaration statement* lists variables of a program, and sometimes indicates the type of data that is stored by the variable. For example, in javascript, `var count, total;` declares two variables, whose types are not defined. In java, `int count;` declares a variable of integer type, and `String phrase;` declares a variable (called `phrase`) of string type. In javascript, `var cutoffs = new Array();` declares `cutoffs` to be an array.

- **Assignment statements.** Assignment statements assign values to variables. Examples are:

```
total = total + count;
text = 'Hello';
a[3] = text;
count += f(3,y);
```

Suppose that we use boxes to represent the memory in which variable data are stored, and write the values of the variables in the boxes. Suppose also that the call `f(3,y)` returns 2 (see more on function calls below). Suppose that *before* the above statements are executed, the variable values are:

```
total: 

|   |
|---|
| 3 |
|---|


count: 

|   |
|---|
| 7 |
|---|


text: 

|  |
|--|
|  |
|--|


a: 

|     |      |        |        |
|-----|------|--------|--------|
| bug | moth | insect | phrase |
|-----|------|--------|--------|


```

Then, upon execution of the above assignment statements, the values would be:

```
total: 

|    |
|----|
| 10 |
|----|


count: 

|   |
|---|
| 9 |
|---|


text: 

|       |
|-------|
| Hello |
|-------|


a: 

|     |      |       |        |
|-----|------|-------|--------|
| bug | moth | Hello | phrase |
|-----|------|-------|--------|


```

Sometimes, assignment statements are used within a variable declaration statement to provide an initial value for a variable:

```
var total, count = 0;
var answers = new Array(''(unanswered)'');
var cutoffs = new Array(50,60,70,80);
```

If we use boxes to represent the memory in which variable data are stored, then upon execution of the above variable declaration statements, we would have the following representation:

```
total:  
count:  
answers: 
cutoffs:    
```

- **If statements.** These are either of the form:

```
if (condition) { statements }
```

or

```
if (condition)
  { statements }
else
  { statements }
```

The condition might be, for example, `total == 10`, in which case the condition would be *true* if the value stored in variable `total` were 10 and would be *false* otherwise. Another example would be the condition `text == 'good'`. If the condition is true, the statements in brackets following the condition are executed, otherwise the statements in brackets after the `else` are executed, if any.

- **While loop statements.** These are of the form:

```
while (condition)
  {statements}
```

The statements are executed over and over again, while the condition is true.

- Additional statements are used to read data from the user and write out data for the user to view (details omitted here).
- **Functions (or methods).** A function takes data as parameters, and returns a result that is calculated from the data. The general form, and an example, are as follows:

```
function f(a,b)
{
  ... statements ...
  return variable
}
```

```
function f(a,b)
{
  var c;
  c = a + b;
  return c;
}
```

The function may be *called* as part of an assignment statement or condition. For example, the statement `count += f(3,y);` is an assignment statement that calls a function `f` which has two parameters. When the call is made the values of the parameters in the call are *passed* to the variables listed at the line where the function is declared. In the example above, if `y` has value 5, then the value 3 is passed to variable `a`, and the value 5 is passed to variable `b`. So, when the function executes, `a` has value 3 and `b` has value 5. Then, `c` is assigned the value 8, and this value is returned by the function and added to `count`.

2 What happens when code executes?

When code is executed, statements are followed in order from “start” to “end”. Data values are created and changed; typically some of these data values are supplied by the user of the code. Control flow statements such as `if` and `while` loops allow some control over the order in which instructions are executed. Upon a function call, statements in the function are executed and then control passes back to the point at which the function is called. Some statements provide output to the user (images, or text, for example).

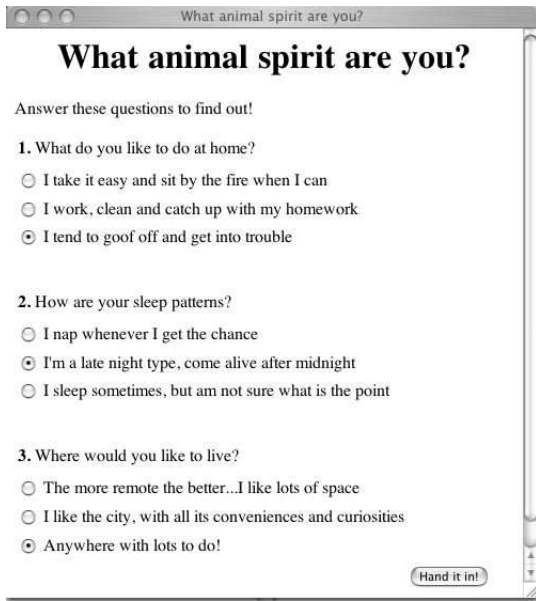
A good way to understand what happens when code is executed is to keep track of the data values stored in the variables as the statements are executed. Follow the statements from the point at which the code starts, and at each statement, update the values of the data items.

3 Exercise

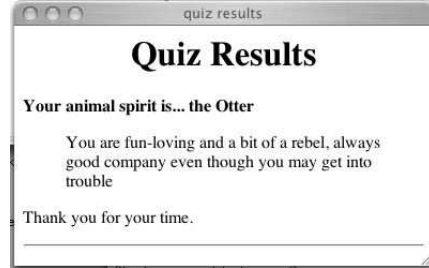
Following is code for a quiz, which you saw earlier this semester. The quiz is stored in file `quiz.htm`, and the results are presented by file `quiz-results.htm`. Can you identify components of the code listed above? Can you trace out what happens when the code executes? Figures 1 (a) and (b) below show you the user’s view of the web pages for the corresponding quiz, and the results, as presented by a browser.

This quiz has three questions, each with three choices. As with the quiz you worked with in the lab, the user selects one answer per question, by clicking a radio button. For each question, one answer has value 0, one has value 1 and one has value 2, and the output that the user sees depends on their total score.

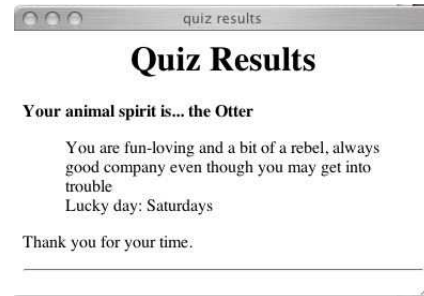
1. The code in the *body* of this quiz has only one variable. What is the name (i.e. identifier) of this variable?
2. At which line of the code is this variable declared?



(a)



(b)



(c)

Figure 1: (a) User view of quiz web page, with certain answers chosen. (b) Output that is presented to the user, when the button marked “Hand it in” is clicked on the page in Figure 1(a). (c) You will need to refer to this figure for one of the questions on page 2.

3. Suppose that when you try the quiz, you choose answers for questions 1, 2, and 3 that happen to have values 2, 1, and 2, respectively. In this case, for each row of the table below, fill the value that `score` would have, just after the code line given in that row has executed.

code line	value of score
30	
31	
32	
33	
34	
40	

4. Again, suppose that you choose answers for questions 1, 2, and 3 that have values 2, 1, and 2, respectively. What animal would be presented to the user by the `quiz-result.htm` page?
5. Once again, suppose that you choose answers for questions 1, 2, and 3 that have values 2, 1, and 2, respectively. Now, suppose also that line 33 of the code is changed to be:

```
33. score += handleQuestion(2);
```

That is, the parameter passed to the `handleQuestion` function on line 33 is 2, not 3. In this case, what animal do you think would be output?

6. Suppose that, when the Otter is output, we want the quiz results page to look like Figure 1 (c) rather than Figure 1 (b). How would you change the code to make this happen?

4 quiz-result.htm file

```
1. <html>
2. <head>
3. <title>quiz results</title>
4. <script language=''JavaScript1.1''>
5. <!--
6.     var lString = document.location + '';
7.
8. function handleQuestion(number) {
9.     var i, j, userAnswerNumber;

    /* This function calculates the value of the user's answer to a question whose
       number is passed as a parameter. The value may be 0,1, or 2. If the user
       does not answer the question, the value is set to 0. The value is returned
       by the function.
    */

10.    if((i = lString.indexOf(number-1 + '=')) >= 0) {
11.        i = lString.indexOf('=', i);
12.        j = lString.indexOf('&', i);
13.    }
14.    if(i < 0 || j < 0) {
15.        userAnswerNumber = 0;
16.    }
17.    else {
18.        userAnswerNumber = parseInt(lString.substring(i + 1, j));
19.    }
20.    return userAnswerNumber;
21. } // end handleQuestion()
22.
23. //-->
24. </script>
25. </head>

26. <body>
27. <h1 align=center>Quiz Results</h1>
28. <script language=''JavaScript1.1''>
29. <!--
```

```

30. var score = 0;

31. score += handleQuestion(1);
32. score += handleQuestion(2);
33. score += handleQuestion(3);

34. if (score < 3) {
35.     document.writeln('<<p><strong> Your animal spirit is... the Bear </p> </strong>');
36.     document.writeln('<<blockquote>');
37.     document.writeln('You are patient, but stubborn; big-hearted and lovable');
38.     document.writeln('<</blockquote>');
39. }
40. else if (score > 4) {
41.     document.writeln('<<p><strong> Your animal spirit is... the Otter </p> </strong>');
42.     document.writeln('<<blockquote>');
43.     document.writeln('You are fun-loving and a bit of a rebel, always good company even though you may get into trouble');
44.     document.writeln('<</blockquote>');
45. }
46. else {
47.     document.writeln('<<p><strong> Your animal spirit is... the Horse </p> </strong>');
48.     document.writeln('<<blockquote>');
49.     document.writeln('You are caring and free-spirited, and are well loved');
50.     document.writeln('<</blockquote>');
51. }
52. //-->
53. </script>
54. Thank you for your time.
55. <p>
56. <hr>
57. </body>
58. </html>

```

5 quiz.htm file

You should be able to answer the above questions without looking at this code, but it is presented here in case you want to check it out.

```

<html>
<head>
<title>What animal spirit are you?</title>
</head>
<body bgcolor=#ffffff>

```

```
<h1 align=center>What animal spirit are you?</h1>
Answer these questions to find out!
<p>
<form name=quizquiz method=get action=quizquizresult.html>
<table height='438'>

<tr>
  <td valign=top height='25'><strong>1.</strong>
  What do you like to do at home? </td>
</tr>
<tr>
  <td height='24'><input type=radio name=0 value=0> I take it easy and sit by the
    fire when I can </td>
</tr>
<tr>
  <td height='24'><input type=radio name=0 value=1> I work, clean and catch up with
    my homework </td>
</tr>
<tr>
  <td height='24'><input type=radio name=0 value=2> I tend to goof off and
    get into trouble <br><br><br></td>
</tr>
<tr>
  <td valign=top height='25'><strong>2.</strong>
  How are your sleep patterns? </td>
</tr>
<tr>
  <td height='24'><input type=radio name=1 value=0> I nap whenever I get the chance</td>
</tr>
<tr>
  <td height='24'><input type=radio name=1 value=2> I'm a late night type,
    come alive after midnight</td>
</tr>
<tr>
  <td height='24'><input type=radio name=1 value=1> I sleep sometimes,
    but am not sure what is the point<br><br><br></td>
</tr>
<tr>
  <td valign=top height='25'><strong>3.</strong>
  Where would you like to live? </td>
</tr>
<tr>
  <td height='24'><input type=radio name=2 value=1> The more remote the better...
    I like lots of space </td>
```

```
</tr>
<tr>
  <td height='24'><input type=radio name=2 value=0> I like the city, with all
    its conveniences and curiosities </td>
</tr>
<tr>
  <td height='24'><input type=radio name=2 value=2> Anywhere with lots to do!</td>
</tr>
<tr>
  <td height='28'></td>
  <td colspan=3 height='28'><input type=submit value='Hand it in!' name=submit></td>
</tr>
</table>

</form>
<hr>
</body>
</html>
```