W3101 Scripting Languages — Javascript Midterm exam Apr 01, 2009

Name		
Iname.		

Student Id:

1. What is the difference between private members and privileged members of Javascript? Explain how they restrict access and give privileges to internal (data and functional) members and external variables and functions. ... (2 marks)

Solution: A private member in Javascript can be accessed only by private members and privileged methods. They can not be accessed by public members and external functions. Privileged methods can access private members, and can be called by public members and external functions.

- 2. Give the definition for two objects, "employee" and "executive" such that ... (4 marks)
 - (a) employee is the parent and executive is the child.
 - (b) Define a private member called "name" in employee.
 - (c) Define a privileged function called "getSalary()" in employee that returns some constant value s1, which is the salary of the employee. Define another privileged function called "getSalary()" in executive that returns some constant value s2, which is the salary of the executive.
 - (d) Explain precisely how Javascript is enforcing OOP concepts, specifically data encapsulation, inheritances and polymorphism with respect to this example.

Solution:

```
var s1 = 100000;
var s2 = 200000;
function employee()
{
    var name;
    this.getSalary = function() { return (s1); }
}
function executive()
    {
    this.getSalary = function() { return (s2); }
    }
executive.prototype = new employee;
```

Javascript is not a strongly typed language. It is class free, there is no keyword "class". It provides prototypal inheritance and supports data encapsulation using "public", "private" and "privileged" identifiers.

In the above example, "name" is a private member whose access is restricted to private and privileged functions. Javascript supports inheritance — in the above example "executive" is derived from "employee". Javascript supports polymorphism, it calls the functions specific to the object bottom up — that is, it calls a member function in the most specific derived class. If it is not found there, it tries to find the method in the parent class. (Since Javascript is weakly typed, there are no complications found in C++ to implement polymorphism using virtual functions.)

- 3. Answer the following questions briefly and precisely:
 - (a) Javascript events what are they, why are they useful? ... (1 mark)

Solution: Events occur because of some kind of input or errors in web page, etc. In Javascript event are defined for web page loading or unloading, mouse button click, key press, etc. They are useful to associate functions with different events, that is, calling a particular function when a mouse button is pressed, or key is pressed, web page is loaded or unloaded, etc. This can help in animation, and many other features of web programming.

- (b) List at least 6 Javascript events and say when they are invoked. ...(1 mark) **Solution:**
 - (i) onclick Mouse clicks an object.
 - (ii) onsubmit Submit button is pressed.
 - (iii) onmousedown The mouse button is pressed.
 - (iv) onmouseup The mouse buttion is released.
 - (v) onkeypress A key is pressed.
 - (vi) onkeyup A key is pressed and released.
 - (vii) onmouseover The mouse is put over an element, like an image.
 - (viii) onmouseout The mouse is taken out of an element, like an image.
- (c) Show how you would use any one of your favorite Javascript events in a small code segment. (If you plan to call a function, you can assume the function exists; you don't have to write the code for the function). ... (1 mark)

Solution: <button onclick="clickFunction()''> The function '`clickFunction()'' is called is called when the button (for which this event is defined) is clicked.

(d) Write a small program that checks if a broswer supports Javascript or not. If it does not support Javascript, it should ask the user to use a different browser. ...(1 mark)

Solution: <noscript> Your browser does not support Javascript! Please use a different browser. </noscript>

4. What does the "===" operator in Javascript do? Is it the same as "==" operator? If not, clarify. What is the output of following code segment: ... (3 marks)

```
var i = 3;
var j = 3;
var k = ``3'';
document.write (``i == j: '' + (i==j));
document.write (``i == k: '' + (i==k));
document.write (``i === j : '' + (i===j));
document.write (``i === k: '' + (i===k));
```

Solution: The "===" operator in Javascript compares both the type and value of two variables. This is unlike the "==" operator, which only compares the value of two variables.

- true true true false
- 5. Write a Javascript program/function that shows the numbers [1-10] in an alert box, one after another sleeping for 100, 200, ..., 900 msecs. That is, when the program starts, it should show "1". Then it should sleep for 100 msec. and show "2", sleep for 200 msec. and show "3" and so on, finally showing "10" before returning. ...(2 marks) **Solution:**

```
var a = 1;
function showNumbers()
{
    alert (a++);
    if (a < 11)
        setTimeout ("showNumbers()", (a-1) * 100);
}</pre>
```