

## COMP 110 Practice Exercises for Midterm – Solutions

Answers are highlighted in blue color.

Evaluate each of the following Javascript expressions and show the value.

1.  $-9 * 3$  -27
2. "value is " + 50 "value is 50"
3.  $17 \% 5$  2
4.  $5 \% 17$  5
5.  $5 / 10$  0.5
6.  $(4 == 4)$  true
7.  $(4 != 5)$  true
8.  $(7 <= 8)$  true
9.  $\text{Math.ceil}(x) - \text{Math.floor}(x)$  1

Convert these binary numbers into decimal numbers.

10. 110111 55
11. 101011 43
12. 10100 20

Convert these decimal values into binary numbers.

13. 55 110111
14. 43 101011
15. 20 10100

Write javascript code to do the following.

16. Alert "Hello world."

```
alert("Hello world");
```

17. Read a number (using prompt) and display it using alert.

```
var n = prompt("Please enter a number.");  
alert("Your number is " + n);
```

18. Read two numbers and display their product.

```
var n1 = prompt("Please enter a number.");
```

```
var n2 = prompt("Please enter another number.");
alert("The product of "+n1+" and "+n2+" is "+ n1*n2);
```

19. Read two numbers and display their sum. What problem did you encounter?

```
var n1 = 1*prompt("Please enter a number.");
var n2 = 1*prompt("Please enter another number.");
alert("The sum of "+n1+" and "+n2+" is "+ n1+n2);
```

If you leave out multiplying the numbers by 1, you will get concatenation instead of addition since `prompt` returns a string.

20. Read in two numbers and display the larger.

```
var n1 = 1*prompt("Please enter a number.");
var n2 = 1*prompt("Please enter another number.");
if (n1 > n2)
    alert(n1+" is the larger.");
else
    alert(n2+" is the larger.");
```

If you leave out the multiplication by 1, you will get strange answers since `prompt` returns a string. For example, while the number 100 is greater than the number 99, the string "99" is greater than the string "100".

21. Read in two numbers and display them in ascending order.

```
var n1 = 1*prompt("Please enter a number.");
var n2 = 1*prompt("Please enter another number.");
if (n1 > n2)
    alert(n2+" "+n1);
else
    alert(n1+" "+n2);
```

22. Use a loop to display the numbers 0 through 5, each in a separate alert window.

```
for (var i=0; i<=5; i++)
{
    alert(i);
}
```

23. Use a loop to display the numbers 0 through 5 in a single alert window.

```

var s=""; // Create empty string.
for (var i=0; i<=5; i++)
{
    s = s + i + " "; // Add next integer plus a space.
}
alert(s);

```

24. Use a loop to display the numbers in the range 0...20 that are multiples of 3.

```

// Solution 1: Generate all numbers and test each.
for (var i=0; i<=20; i++)
{
    if (i%3==0)
        alert(i); // Display those that are
                  // divisible by 3.
}

// Solution 2: Generate only multiples of 3.
for (var i=0; i<=20; i=i+3)
{
    alert(i); // Only multiples of 3 are generated.
}

```

25. Use a loop to display the integers 9 through 0 in descending order.

```

for (var i=9; i>=0; i--)
{ alert(i);}

```

26. Prompt the user for a number in the range 0...100. If the number is out of range, display an error message and prompt again until a valid number is entered. Assume the user enters a number each time.

```

// Solution 1: Put the test in the while statement.
var n = prompt("Please enter a number in the range 0...100");
while (n<0 || n>100)
{
    alert(n + " is out of range. Try again.");
    n = prompt("Please enter a number in the range 0...100");
}

```

```

}

// Solution 2: Put the test inside the loop.
while (true)
{
    var n = prompt("Please enter a number in the range 0...100");
    if (n>=0 && n<=100) break;
    alert(n + " is out of range. Try again.");
}

```

27. Repeat previous exercise, but this time allow for the possibility that the user enters something that is not a number. Hint: the built in function `isNaN(x)` returns **true** if x is not a number. It returns **false** if x is a number. Sort of backwards, but that's what's available.

```

// Solution 1: put the test in the while statement.
var n = prompt("Please enter a number in the range 0...100");
while (isNaN(n) || n<0 || n>100)
{
    alert(n + " is out of range or not a number. Try again.");
    n = prompt("Please enter a number in the range 0...100");
}

// Solution 2: put the test inside the loop.
while (true)
{
    var n = prompt("Please enter a number in the range 0...100");
    if (!isNaN(n) && n>=0 && n<=100) break;
    alert(n + " is out of range or not a number. Try again.");
}

```

28. Prompt for an integer, then display the sum of the integers from 0 through the number entered. For example, if you enter 10, then display 55 which is the sum of 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10.

```

// Get the upper bound.
var max = 1*prompt ("Please enter an integer.");
var sum = 0;
for (var i=0; i<=max; i++)

```

```

        sum = sum + i; // Add i to the running sum.

    alert("The sum of 0 through "+max+" is "+sum);

```

29. Prompt for an integer, then display the average of the integers from 0 through the number entered. For example, if you enter 10, then display 5 which is the average of  $(0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10)/11$ .

```

    // Get the upper bound.

    var max = 1*prompt ("Please enter an integer.");
    var sum = 0;

    for (var i=0; i<=max; i++)

        sum = sum + i; // Add i to the running sum.

    alert("The average of 0 through "+max+" is "+(sum/(max+1)));

```

**Write a function to do each of the following.**

30. `greet()` ; displays "Hello world"

```

function greet()
{
    alert("Hello world");
}

```

31. `sum(n)` ; displays the sum of  $0+1+2+...+n$ .

```

    // Sum the integers 0...n.

    function sum(n)
    {
        var tot=0;

        for (var i=0;i<=n;i++)

            tot=tot+i;

        alert("The sum of 0 through "+n+" is "+tot);
    }

```

32. `isValid(n)` returns true if n is a number 0...100.

```

    // Is the parameter between 0 and 100?

    function isValid(n)
    {
        // Restriction: none

        // Errors checked for: n not a number;

```

```

//                      n out of range.
if (isNaN(n)) return false; // Not a number.
return n>=0 && n<=100;      // Range test.
}

```

33. `isInteger(n)` returns true if `n` is an integer 0...100. Hint: use `isValid`.

```

// Is n an integer in 0 ... 100?
function isInteger(n)
{
    // Restriction: none
    // Errors checked for: n not a number; n out of range;
    //                      n not an integer.
    if (!isValid(n)) return false // Not a number or out of range.

    // If we get to this point, n must be a number in range.
    return (Math.floor(n)==n)      // Integer test.
}

```

#### Code tracing:

34. Trace the following code by showing the values of the 3 variables in the table on the right, for each line of code that is executed (*after* the line is executed):

```

// Start of execution
var x = 5;      ----->
var y = 10;     ----->
var z = 7;      ----->
x = (y+z)/2;    ----->
y = 8;          ----->
z = (x-y)/2;    ----->

```

| x   | y         | z         |
|-----|-----------|-----------|
| 5   | Undefined | Undefined |
| 5   | 10        | Undefined |
| 5   | 10        | 7         |
| 8.5 | 10        | 7         |
| 8.5 | 8         | 7         |
| 8.5 | 8         | 0.25      |

35. Trace the following program by showing the text of the alert messages that are displayed when it runs. There are no errors.

```

// Start of execution

```

```

var x =5;

var y =1;

while (x > 0){

    x = x-1;

    y = y*x;

    alert(x + " " + y);

}

```

```

4 4
3 12
2 24
1 24
0 0

```

```

var i;

var count =0;

for (i =0; i < 11; i++){

    if (i < 3 || 7 <i){

        count++;

        alert(count + " i " + i + " range 1");

    }else if (i == 5){

        count++;

        alert(count + " i " + i + " range 2");

    }

    if ( 2<= i && i < 7){

        count++;

        alert(count + " i " + i + " range 3");

    }

}

```

```

1 i 0 range 1
2 i 1 range 1
3 i 2 range 1
4 i 2 range 3
5 i 3 range 3

```

6 i 4 range 3  
 7 i 5 range 2  
 8 i 5 range 3  
 9 i 6 range 3  
 10 i 8 range 1  
 11 i 9 range 1  
 12 i 10 range 1

36. What is the output displayed by each of the following code fragments?

```
for (var i=0; i<10; i++)
{
    alert(i);
}
```

10 alerts with the values 0...9.

```
for (var i=10; i<10; i++)
{
    alert(i);
}
```

Nothing. The loop condition fails immediately.

```
for (var i=10; i>=0; i--)
{
    alert(i)
}
```

11 alerts with the values 10 down to 0.

```
for (var i=0; i<5; i++)
{
    for (var j=0; j<3; j++)
    {
        alert(i + " " + j);
    }
}
```

15 alerts with the following values.



0 0  
0 1  
0 2  
1 0  
1 1  
1 2  
2 0  
2 1  
2 2  
3 0  
3 1  
3 2  
4 0  
4 1  
4 2

Remember, when one loop is inside another, the inner loop runs to completion for every step of the outer loop.

```
for (var i=0; i<5; i++)  
{  
  for (var j=i; j<5; j++)  
  {  
    alert(i + " " + j);  
  }  
}
```

15 alerts with the following values. Note that the inner loop starts at a different value each time.

0 0  
0 1  
0 2  
0 3  
0 4

1 1  
 1 2  
 1 3  
 1 4  
 2 2  
 2 3  
 2 4  
 3 3  
 3 4  
 4 4

37. What does this code do?

```
var count=0;
for (var half=0; half<=2; half++)
{ for (var qtr=0; qtr<=4; qtr++)
  { for (var dime=0; dime<=10; dime++)
    { for (var nick=0; nick<=20; nick++)
      { for (penny=0; penny<=100; penny++)
        { if (50*half + 25*qtr + 10*dime + 5*nick + penny == 100)
          { count++; }
        }
      }
    }
  }
}
alert(count);
```

It counts and displays the number of different way to make \$1.00 using half dollars, quarters, dimes, nickels, and pennies. It tests 349,965 different combinations to find the 292 combinations that add up to exactly \$1.00.