Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Number : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Spring 2015 CENG 2002 Midterm Exam

Points(120 pts): Q1-Q2: 10pts, Q3-Q6: 15pts, Q7-Q8: 20pts

1. (10 pts) Let’s suppose that in language A every operator is left associative, + and – has the same and lowest precedence and \* has higher precedence than /:

p(\*) > p(/) > p(+) = p(-)

In such a language what will be the value and corresponding expression tree of this expression:

6\*2-4+20/5\*2-10

1. (10 pts) Give the output of the following code segment. Additionally, give the equivalent Python conditional expression at line 2.

var a = 5; b = 7;

var c = a+b>10 ? (a<b ? 10 : 20) : 30; // line 2

var d = ("abc" || 7) && ("" || 7<3 || 0) && 3;

console.log(c,d);

1. (15 pts) What will be output of the following Python code? Note that Python also has closures but if you are going to modify an externally defined variable, you need to tell Python first about this (see the comments):

def f(a,inc):

def g(b):

nonlocal a # we need this in python since we want to modify a

a += inc # inc is not modified so no need to declare it nonlocal

return b\*a

return g

g = f(5,1)

print(g(7))

h = f(3,2)

print(h(7))

print(h(8))

print(h(7))

print(g(7))}

1. (15 pts) Give the output of the following code:

var people = [ {name:'ali', age:19}, {name:'oya', age:7}, {name:'izzet', age:25},

{name:'can', age:93}, {name:'zeynep', age:11} ]

var result = people.reduce(function(tally, nextPerson){

if (nextPerson.age < 20) tally.x++;

else {

tally.y++;

tally.last = nextPerson.name;

}

return tally;

}, {x: 0, y:0) );

console.log(result); // print contents of variable result

1. (15 pts) What will be the output of this code segment:

function f(obj, a, b, c, x, y) {

obj.a++;

a++;

obj.b.push(30)

b.push(30)

b[0]++;

y++;

obj.c.x = "def"

c.x = "def"

x = "def"

}

var obj = { a:1, b:[10,20], c: {x:"abc"} }

var a = 1

var b = [10,20]

var c = {x:"abc"}

var x = "abc"

var y = b[0]

f(obj,a,b,c,x,y)

console.log(obj,a,b,c,x,y) // give each variable’s full value

console.log(c == obj.c)

1. (15 pts) Give the values of each local variable in function f at the end of its execution (i.e. the output of the print statements at the end):

def f():

a = [ 3, {'x':3, 'y':[10,20]}, [1,2,3] ]

# assignments

b = a[:]

c = b

y = a[1]['y']

y1 = y[1]

y2 = a[1]['y'][:]

# modifications

a[2] = [1,2,3]

y1 = 30

y2.append(50)

c[2].append(70)

a[1]['x'] = 90

b[0] = "xyz"

test = y is y2

for (k,v) in locals().items():

print("{0} = {1}".format(k,v))

#

f()

1. (20 pts) What will be the output of this code segment:

var x=1, y=2, z=3;

function f() {

var z = 9;

x = 4;

g(x,y)

console.log("f", x, y, z)

}

function g(a,b) {

var y = 6;

z = 7

function h(z) {

x++; y++; z++

console.log("h", x, y, z);

}

h(z);

console.log("g", x, y, z);

}

console.log("global", x, y, z)

f()

console.log("global", x, y, z)

1. (20 pts) What will be the output of this code segment:

var fs = {

a: function (t1,t2) {

if (t2.name.length != t1.name.length)

return t2.name.length - t1.name.length;

else if (t1.points != t2.points)

return t2.points - t1.points;

else

return t1.played - t2.played;

},

b: function(t) {

console.log(t.name + ": " + t.points + "-" + t.played);

}

}

var g = function(n) {

return function(t) {

return t.played >= n;

}

}

function compute(t) {

t.points = 2\*t.wins+t.draws;

t.played=t.wins+t.draws+t.losses

}

var teams = [

{ name: "HataySpor", "wins": 3, "draws": 2, "losses": 1},

{ name: "HataySporrr", "wins": 2, "draws": 2, "losses": 2},

{ name: "Karabük", "wins": 7, "draws": 2, "losses": 1},

{ name: "Muğlaspor", "wins": 3, "draws": 2, "losses": 0},

{ name: "UlaGücü", "wins": 1, "draws": 3, "losses": 0},

{ name: "Kötekli", "wins": 1, "draws": 7, "losses": 0},

{ name: "Dalaman", "wins": 2, "draws": 0, "losses": 0}

]

teams.map(compute)

teams.filter(g(3)).sort(fs.a).forEach(fs.b)