

Second Midterm Exam

CS164, Fall 2007

Nov 8, 2007

- Please read all instructions (including these) carefully.
- Write your name, login, and SID.
- No electronic devices are allowed, including cell phones used as watches.
- Silence your cell phones and place them in your bag.
- The exam is closed book, but you may refer to one (1) page of handwritten notes.
- Solutions will be graded on correctness and **clarity**. Each problem has a relatively simple and straightforward solution. Partial solutions will be graded for partial credit.
- There are 6 pages in this exam and 4 questions, each with multiple parts. If you get stuck on a question move on and come back to it later.
- You have 1 hour and 20 minutes to work on the exam.
- Please write your answers in the space provided on the exam, and clearly mark your solutions. You may use the backs of the exam pages as scratch paper. **Do not** use any additional scratch paper.

LOGIN: _____

NAME: _____

SID: _____

Problem	Max points	Points
1	20	
2	30	
3	30	
4	20	
TOTAL	100	

Problem 2: Visitors, Object-Orientated Programming [30 points]

```

interface ASTNode {
    public void accept (Visitor v);
}

class IdentifierNode implements ASTNode {
    public final String id;
    public IdentifierNode (String id) {
        this.id = id;
    }
    public void accept (Visitor v) {
        System.out.println("1");
        v.visit(this);
    }
}

class NumberNode implements ASTNode {
    public final String num;
    public NumberNode (String num) {
        this.num = num;
    }
    public void accept (Visitor v) {
        System.out.println("2");
        v.visit(this);
    }
}

class BinopNode implements ASTNode {
    public final String op;
    public final ASTNode lhs;
    public final ASTNode rhs;
    public BinopNode (ASTNode lhs,
        String op,
        ASTNode rhs) {
        this.lhs = lhs;
        this.op = op;
        this.rhs = rhs;
    }
    public void accept (Visitor v) {
        System.out.println("3");
        v.visit(this);
    }
}

class Visitor {
    public void visit (ASTNode n) {
        System.out.println("4");
        n.accept(this);
    }
    public void visit (IdentifierNode n) {
        System.out.println("5");
    }
    public void visit (NumberNode n) {
        System.out.println("6");
    }
    public void visit (BinopNode n) {
        System.out.println("7");
        n.lhs.accept(this);
        n.rhs.accept(this);
    }
}

class FindFree extends Visitor {
    public Set variables;
    public FindFree () {
        this.variables = new HashSet();
    }
    public void visit (IdentifierNode n) {
        System.out.println("8");
        variables.add(n);
    }
}

// Construct AST for expression 1+i
ASTNode a = (ASTNode) new BinopNode(
    new NumberNode("1"),
    "+",
    new IdentifierNode("i")
);

Visitor v = (Visitor) new FindFree();
v.visit(a);

```

Part 1 [15 points]: What does this program output? **Answer:** _____

Part 2 [7 points]: Circle one call expression that must decide at *run time* which of multiple functions to call. Indicate among what functions it is choosing, by drawing arrows from the call to the function definitions.

Part 3 [8 points]: Put in a box one call expression at which the compiler must decide at *compile time* which of multiple functions to call. Indicate among what functions it is choosing, by drawing arrows from the call to the function definitions.

Problem 3: Closures [30 points]**Part 1 [5 points]:** What does the following program output? **Answer:** __, __, __, __

```
function f()
  local x = 1;
  return function () x = x+1; return x end
end

i1 = f()
i2 = f()
print(i1(), i2(), i1(), i2())
```

Part 2 [5 points]: How many copies of variable x are created during the execution of the above program? **Answer:** __**Part 3 [5 points]:** Indicate (by drawing a '*' sign) where in the code can these variable(s) be deallocated. Indicate the *earliest* point(s) where deallocation is safe.**Continues on next page**

Part 4 [15 points]: The Lua code below implements a list module (you have seen a similar code in the lecture). Your goal is to implement the function `iterator()` so that it can be used to iterate over the values of the list exactly as shown below.

Use closures to implement the iterator functionality. Assume that the list is not modified while an iterator operates on it, but two iterators may simultaneously iterate over the same list.

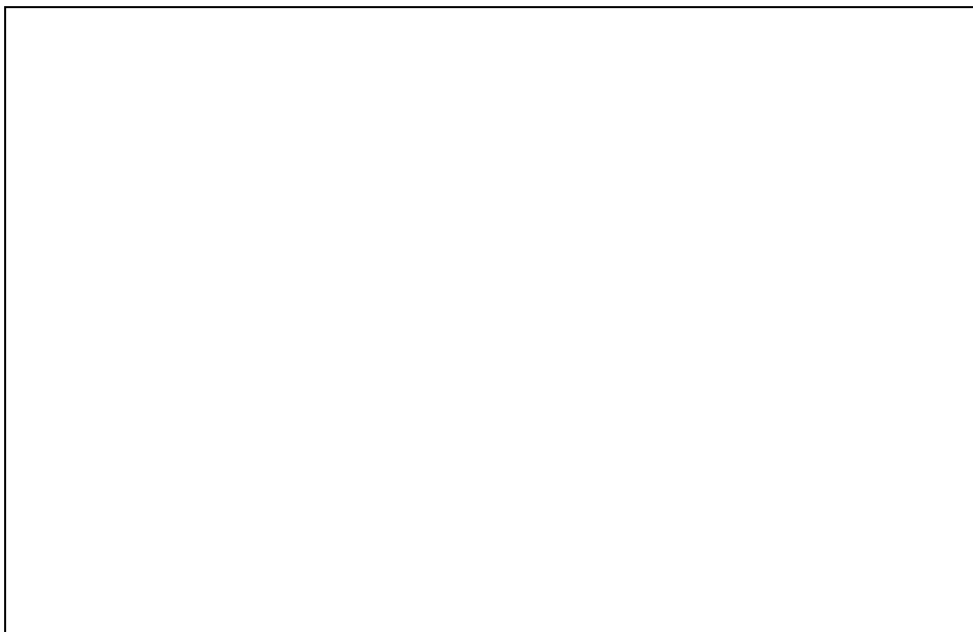
```
List = {}
function List.new ()
    return {first=0, last=0}
end
function List.push(list,value)
    list[list.first] = value
    list.first = list.first+1
end
function List.remove(list,value)
    if list.last >= list.first then return nil end
    local last = list.last
    list.last = list.last + 1
    return list[last]
end

myList = List.new()

List.push(myList, 1)
List.push(myList, 2)
List.push(myList, 3)

next = List.iterator(myList)

while true do
    element = next()
    if element == nil then break end
    print(element)    -- outputs 1 2 3
end
```



Problem 4: Syntax Directed Translation [20 points]

The code below implements a simple version of your Google calculator from Project 3. You are asked to find and correct a bug in the code. The incorrect behavior can be seen on the input **-3!**. The correct output should be **-6**. However, the calculator outputs an error message. *Hint*: the bug is not in the factorial() function.

```
%import mymodule
%ignore /[ \t\n]/
%left '+'
%right '-'
%left '!'
%%
S -> E                %{ print(n1.val); %}
;
E -> E '+' E          %{ return(n1.val+n3.val); %}
| E '-' E             %{ return(-n2.val); %}
| E '!'              %{ return(mymodule.factorial(n1.val)); %}
| /(-)?(\d+\.?\d*|\d+|\.\d+)/  %{ return float(n1.val); %}
;
```

Mymodule.py:

```
import math

def factorial(n):
    if n < 0:
        print 'factorial is undefined for negative numbers'
        raise Exception()
    if math.modf(n)[0] != 0:
        print 'factorial is undefined for non-integers'
        raise Exception()
    result = 1
    for i in range(1,int(n)+1):
        result *= i
    return result
```

Part 1 [13 points]: Explain precisely why the calculator behaves incorrectly. For full credit, you must explain what went wrong on the input **-3!**. Draw a figure illustrating the problem on this input.

Part 2 [7 points]: Correct the bug. Make the change directly in the given code.