CMSC 330             Exam #2             Spring 2006

Do not open this exam until you are told. Read these instructions:

1. This is a closed book exam. No notes or other aids are allowed. If you have a question during the exam, please raise your hand. Each question’s point value is next to its number.

2. You must turn in your exam immediately when time is called at the end.

3. 6 pages, 5 problems, 100 points, 50 minutes.

4. In order to be eligible for as much partial credit as possible, show all of your work for each problem, and clearly indicate your answers. Credit cannot be given for illegible answers.

5. You will lose credit if any information above is incorrect or missing, or your name is missing from any side of any page.

6. Parts of this page and of two other pages are for scratch work. If you need extra scratch paper after you have filled these areas up, please raise your hand. Scratch paper must be turned in with your exam, with your name and ID number written on it. Scratch paper will not be graded.

7. To avoid distracting others, no one may leave until the exam is over.

8. The Campus Senate has adopted a policy asking students to include the following handwritten statement on each examination and assignment in every course: “I pledge on my honor that I have not given or received any unauthorized assistance on this examination.” Therefore, just before turning in your exam, you are requested to write this pledge in full and sign it below:

______________________________________________________________________________

Good luck!

SCRATCH WORK AREA
(this area will not be graded)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. [26 pts.] Short-answer questions based on material discussed in class— for those which require explanation, answer them quickly and very briefly, each in two sentences or less.

a. [8 pts.] On the left below is the the operational semantics rule for function application in Scheme which was given in class. On the right is a version with only one difference (in bold so it stands out). Note: in the lecture slides Scheme program text was shown in blue and values were in red, but Scheme program text in the rules below is shown in normal font, and values are in italic.

\[
\begin{align*}
E; S_1 &\rightarrow (E', \lambda id.S) & E; S_2 &\rightarrow v \\
E; E', id:v; S &\rightarrow v' \\
\hline 
E; (S_1 S_2) &\rightarrow v'
\end{align*}
\]

Given the revised rule, what would be the result of the following Scheme function application?

```
(((lambda (x) (lambda (x) (+ x x))) 3) 4)
```

Briefly explain your answer. (Answers with incorrect or missing explanations won’t receive credit.)

b. [10 pts.] Simplify each of the following lambda calculus expressions:

i. \((\lambda x.\lambda y.x y) z\)

ii. \((\lambda x.\lambda y.x y) y\)

iii. \((\lambda x.\lambda y.x) (\lambda x.x) (\lambda x.x) \lambda x.\lambda y.y\)


c. [8 pts.] Java uses translation via erasure to implement generics. C++ has templates, which are similar to Java’s generics, but C++ can’t use translation via erasure to implement templates. Briefly explain why not.
2. [24 pts.] Suppose you dropped and broke your cell phone, so you don’t have your contact or phone list any more. However, you do have the OCaml interpreter downloaded onto your computer. Write an OCaml function which can be used to create a contact list, for later lookup:

- Your function should be called `create_contact_list` and it should have one parameter, a list of strings. The list consists of a sequence of strings, alternating between your friends’ first names (in no particular order) and their phone numbers.

- Your function `create_contact_list` should **create and return a function** which can then be used to look up your friends’ phone numbers, given their first names. For instance:

```
let lookup_friends = create_contact_list ["Martin"; "(301) 123-4567";
   "Guilherme"; "(301) 111-2222";
   "Asad"; "(301) 987-6543";
   "Denis"; "(301) 222-3333"];;

lookup_friends "Asad";;
- : string = "(301) 987-6543"
lookup_friends "Martin";;
- : string = "(301) 123-4567"
```

- You may assume the list passed into `create_contact_list` will be valid and consist of alternating names and phone numbers as described. You may assume the function it returns will only be called with names which are in the list, and that your friends all have unique first names.

It doesn’t matter if your function would cause incomplete match warnings. To receive full credit, your solution **should not use any references**.
3. [10 pts.] Consider the following program, written in C synax:

```c
#include <stdio.h>

int x = 1;

int f(int a, int b) {
    int c = a + x;
    x += 1;
    x += b + 2;
    return a + c;
}

int main() {
    int y = 7;
    y = f(x + 1, y + x);
    printf("%d %d\n", x, y);
    return 0;
}
```

What would the program print if C used call-by-name as its default parameter transmission method?
4. [25 pts.] Write a context–free grammar which generates the following language:

\[ \{ a^m b^n \mid m + n \text{ is not a multiple of 3} \} \]

For full credit, your grammar should be unambiguous. In order that your answer can be graded as accurately and quickly as possible, please use consecutive nonterminals beginning with S (S, T, U, etc.) when writing your grammar.
5. [15 pts.] Java, subtyping, and generics.

a. [10 pts.] Consider the following Java program:

```java
import java.io.*;
import java.util.*;

class A {
    void g() { System.out.println("g1"); }
}

class B extends A {
    void g() { System.out.println("g2"); }
}

class C {
    static void f(A a) { System.out.println("f1"); }
    static void f(B b) { System.out.println("f2"); }

    public static void main(String args[]) {
        A x = new A();
        A y = new B();
        B z = new B();

        f(x);
        f(y);
        f(z);
    }
}
```

Give the program's complete output below:

```

b. [3 pts.] Suppose the following method is added to class C:

```java
static <T extends B> void h(T t) {
    System.out.println("h");
}
```

Which, if any, of x, y, and z could be passed into h?  

```

c. [2 pts.] What kind of polymorphism are Java's generics an example of?

```