

CMSC330 Fall 2009 Midterm #2

Name _____

Discussion Time (circle one): 10am 11am 12pm 1pm 2pm 3pm

Do not start this exam until you are told to do so!

Instructions

- You have 75 minutes to take this midterm.
- This exam has a total of 100 points, so allocate 45 seconds for each point.
- This is a closed book exam. No notes or other aids are allowed.
- If you have a question, please raise your hand and wait for the instructor.
- Answer essay questions concisely using 2-3 sentences. Longer answers are not necessary and a penalty may be applied.
- In order to be eligible for partial credit, show all of your work and clearly indicate your answers.
- Write neatly. Credit cannot be given for illegible answers.

	Problem	Score
1	Programming Languages	/6
2	Scoping	/8
3	Parsing	/12
4	Lambda Calculus	/16
5	OCaml Types & Type Inference	/8
6	OCaml Programming	/50
	Total	/100

1. (6 pts) Programming languages

a. (3 pts) Describe one design choice for *type declarations* for static types in a programming language, and list a programming language using this approach.

b. (3 pts) How can programmers write Java programs which (effectively) pass functions as arguments to other functions? Give a brief answer.

2. (8 pts) Scoping

Consider the following OCaml code.

```
let app f y = let y = 5 in let x = 7 in let a = 9 in f y ;;  
let add x y = let incr a = a+y in app incr x ;;  
(add 2 4) ;;
```

a. (2 pts) List the order the functions *add*, *incr*, and *app* are invoked in (add 2 4)

b. (3 pts) What value is returned by (add 2 4) with static scoping? Explain.

c. (3 pts) What value is returned by (add 2 4) with dynamic scoping? Explain.

3. (12 pts) Parsing

a. (5 pts) Compute First sets for S and A in the following grammar:

$$\begin{array}{ll} S \rightarrow Acdc & A \rightarrow bgS \\ S \rightarrow aAf & A \rightarrow \varepsilon \quad (* \text{ epsilon } *) \end{array}$$

b. (3 pts) Apply the algorithm discussed in class to transform the following grammar so that it can be parsed using a recursive descent parser.

$$\begin{array}{l} S \rightarrow Sb \\ S \rightarrow ac \end{array}$$

c. (4 pts) Recursive Descent Parsing

Using pseudocode, write a recursive descent parser for the following grammar.

$$\begin{array}{l} S \rightarrow cbS \\ S \rightarrow \varepsilon \quad (* \text{ epsilon } *) \end{array}$$

Use the following utilities:

lookahead	Variable holding next terminal Lookahead == "\$" when at end of input
match (x)	Function to match next terminal to x
error ()	Reports parse error for input

parse_S() {