# CMSC330 Fall 2009 Quiz #2

**Name** _____

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

**Instructions**

- You have 20 minutes for this quiz.
- This is a closed book exam. No notes or other aids are allowed.
- For partial credit, show all of your work and clearly indicate your answers.
- Write neatly. Credit cannot be given for illegible answers.

1. (6 pts) OCaml Types and Type Inference
   a. (2 pts) Give the type of the following OCaml expression

      fun x -> [ x+2 ] **Type =**

   b. (2 pts) Write an OCaml expression with the following type

      int -> bool **Code =**

   c. (2 pts) Give the value of the following OCaml expression. If an error exists, describe the error.

      let x = (fun z -> z – 1) in x 1 **Value =**

2. (6 pts) OCaml Programming

   Using fold and an anonymous function, write a function *numAdults* which when applied to a list of ints *lst,* returns the number of elements of *lst* that are 18 or over. Example: numAdults [17;18; 21;16; 25] = 3

```
let rec fold f a l = match l with
        [] -> a
      | (h::t) -> fold f (f a h) t
```

3. (5 pts) First Sets

Compute First sets for S and A in the following grammar:

$S \rightarrow Ab$ $\quad\quad$ $A \rightarrow dA$

$S \rightarrow c$ $\quad\quad$ $A \rightarrow \varepsilon$ $\quad$ (* epsilon *)

4. (3 pts) Parsing

Finish writing a predictive, recursive descent parser for the following grammar.

$S \rightarrow aSb$

$S \rightarrow \varepsilon$ (* epsilon *)

You may use the following utilities

| | |
|---|---|
| lookahead | Variable holding next terminal |
| match ( x ) | Function to match next terminal to x |
| error ( ) | Reports parse error for input |

```
parse_S( ) {
        if (lookahead == "a") {
                                                    // your code here




        } else {
                                                    // your code here




        }
}
```