

CMSC330 Summer 2009

Quiz 2

Name: _____ **Answers** _____

1. Functional languages (4 points)
 - a. Why is Ocaml considered a functional programming language?

Focuses on expression evaluation with less emphasis on memory writes; has higher-order functions; heavy use of recursion, etc.

- b. Give one reason why someone might want to use a functional language over an imperative one. Your answer should **not** be along the lines "...because they are better at/like functional languages more..."

Easier to parallelize code when there aren't memory writes

2. What is a higher order function? That is, what can you do with higher order functions that you cannot do with "regular" functions? (2 points)

First class variable; can be passed to or returned from a function

3. Give the types (as Ocaml would) of the following expressions, or ERROR if there is an error: (7 points)

a. `[2; 'c']` _____ **error – heterogenous types** _____

b. `(2, 'c')` _____ **int * char** _____

c. `[1;2;3]` _____ **int list** _____

d. `('c'; "hello")` _____ **error – tuple uses , not ;** _____

e. `[[1;2;2];[2;3]]` _____ **int list list** _____

f. `[[[1;2]];[2;3]]` _____ **error – not same level of nesting** _____

g. `Printf.printf "%d" 3` _____ **unit** _____

4. Give the type of the following functions or ERROR if there is an error (4 points):

a. `let head l a = match l with
 [] -> (1, (1+a)) |
 _ -> (0,1)
;;`

_____ **'a -> int -> int * int** _____

b. `let head l = match l with
 [] -> [] |
 _ -> (0,1)
;;`

_____ **error – does not return same type** _____

5. Give the value of the following Ocaml expressions or ERROR if there is an error (5 points)

a. `let x = 3 in
 x::[]` _____ **[3]** _____

b. `let x = 3 in
 x@[1;2;3]` _____ **error – concat list operator expects 2 lists** _____

c. `List.map((fun z -> z + z), [1;2;3])`

d. `let x = 2 in` `[2;4;6]`
`let y = 3 in`
`x::y::[2;4;6]` `[2;3;2;4;6]`

e. `let list = [1;2;3] in`
`match list with`
 `[] -> -1 |`
 `(h::t) -> if (h <> 3) then 0 else 1`
`0`