CMSC330 Fall 2015 Quiz #2 SOLUTIONS

Name: _____

Discussion Time:	**10am**	**11am**	**12pm**	**1pm**	**2pm**	**3pm**
TA Name (Circle):	**Michael**	**Amelia**	**Chris**	**Chris**	**Michael**	**Candice**
	Amelia	**Maria**	**Samuel**	**Josh**	**Max**	

Instructions:

- Do not start this test until you are told to do so!
- You have 20 minutes for this quiz.
- This is a closed book exam. No notes or other aids are allowed.
- Answer essay questions concisely in 2-3 sentences. Longer answers are not needed.
- For partial credit, show all of your work and clearly indicate your answers.
- Write neatly. Credit cannot be given for illegible answers.

1. (4 pts) Give the types of the following OCaml expression

   a. (2 pts) `[[1.0];[2.0;3.0]]` **`Type = float list list`**

   b. (2 pts) `let f (x::_) = x;;` **`Type = 'a list -> 'a`**

2. (3 pts) Write an expression of type `int -> int -> int`

```
let f x y = x+y;;
```

3. (4 pts) Write a recursive function *sumSmall* which takes in an int list *lst* and an integer threshold *x* and recursively sums up the elements of *lst* which are strictly less than *x*. For instance, given the list [1;2;1;4;2;3] 3, *sumSmall* will return 6.

```
let rec sumSmall lst x = match lst with
    (h::t) -> if (h<x) then h + (sumSmall t x) else (sumSmall t x)
  | [] -> 0
;;
```

4. (4 pts) Using map or fold and an anonymous function, write an Ocaml function *timesThree*, which takes in a list of floats *lst* and returns a list of floats in which each element is 3 times greater. For instance, calling *timesThree* on [1.0; 2.0; 3.0] would return [3.0; 6.0; 9.0]. If you do not use map or fold, you will not receive credit.

```
let timesThree lst = map (fun x -> x *. 3.0) lst;;
```