Department of Computer Science and Engineering
University of California, San Diego

CSE 130
Winter 2001

INDIVIDUAL ASSIGNMENT 2

(**due**: Problem 1: Wednesday **Jan. 24**; Problem 2:[1] Monday **Jan. 29**, both **before class**)

- This is **not** a programming assignment. For this individual assignment prepare a *concise* (and readable) answer sheet and hand it on the due date, just **before class**. Write your **name** and **email** address on the front page.

Problem 1 (Types, Function Signatures) Below the function signatures are in the style of Haskell type declarations.

a) Explain the type of each of the expressions $\texttt{f}_1$, $\texttt{f}_2$, and $\texttt{f}_3$:

```
f1 :: String -> Integer
f2 :: Integer -> String
f3 :: (Integer, String) -> Bool
```

Let's assume the *domain* of each $\texttt{f}_i$ is *finite*. What other names (from the class) describe best the type of each $\texttt{f}_i$?

b) Explain the function signatures for

```
f :: a -> (b -> c)
g :: (a -> b) -> c -> d
```

Hint: e.g., for `f` think of integer addition, i.e., let `a = b = c = Integer`. What are the values and types say for `f 17 3` and `f 17` if `f` denotes integer addition.

For `g`, think of the `map` function explained in class. What types make sense for `c` and `d` in case of the `map` function?

Problem 2 (Functions vs Relations) "`(A -> B)`" is the *set of all functions* from A to B. "`(A x B)`" is the *Cartesian product* of A and B. Let $A = \{1, 2, \ldots, n\}$ and $B = \{1, 2, \ldots, m\}$

a) How many elements has "`(A x B)`"?

b) How many elements has "`(A -> B)`"?

c) Let k be given. How can we model a relation over $A_1, \ldots, A_k$ as a function? (Hint: it is sufficient to give a Haskell signature)

[1]But preferably you should turn in Problem 2 together with Problem 1, since there will be a programming assignment coming out on Wednesday!