CMSC 22610 **Computer Language Implementation** **Homework 4**
Winter 2004 **Due February 27**

1. Recall the discussion of polymorphic typechecking in Handout 4. Assume that we have both `int` and `real` as base types. How might we extend the typechecker to support overloaded functions on integers and reals (*e.g.*, "+")? You may assume that such functions are predefined, but you should explain how to modify the representation of types and the typechecking algorithm to handle overloaded functions.

2. Consider the following code in a lexically scoped language:

```
let x = 1;
fun f (x, y) {
  let y = x+y;
  fun g (z) { x+y+z };
  g (y)
}
```

   (a) Rewrite this code adding subscripts (*e.g.*, x_1) to distinguish the different variables.

   (b) What are the free variables of `f` and `g`?