

# Midterm 1, October 15<sup>th</sup> 2001

## CSCI 5535

This is a closed-book exam. You have 50 minutes to answer these questions. We recommend that you read all questions before starting. The points for a question are evenly divided amongst its subparts.

- [25 points]** Consider the type, T, and two functions, f and g, that use the type:

```
type T = {a: Int}
val f = fun (x: T): T = x
val g = all[t? T] fun (x: t): t = x
```

Both these functions can operate on type T and any subtype of T. (i) Are these two functions identical with respect to the value and type that they return? If they are different, explain how they are different. (ii) Are these functions easy to write in your favorite object-oriented language? Why or why not?
- [25 points]** Java's type system allows a narrowing conversion from type T[] (i.e., array of T) to S[] (i.e., array of S) when S and T are reference types and there is a narrowing conversion from T to S. (i) Give an example of this conversion. (ii) Explain why this conversion is allowed only for reference types. (iii) Does the compiler need to insert any run-time checks to support this conversion. If so, give an example when a run-time check is needed. (hint: the run-time checks may happen at the time of the conversion or at some later time).
- [25 points]** In Modula-3 programs you can write `ISTYPE(x, T)` which returns **true** at run time if x points to an object of type T (or subtype of T) and **false** otherwise. (i) Explain how to implement this run-time test. (ii) Does this implementation work in a multiple-inheritance language such as C++? Illustrate with an example.
- [25 points]** Consider the following inheritance hierarchy in C++:

```
class A {int i; };
class B: A {int j;};
class C: A {int k;};
class D: B, C {int n; }
```

i.e., B and C inherit from A and add an instance variable of their own. D inherits from both B and C and adds an instance variable of its own. (i) Show what an object of type D looks like in memory. (ii) Explain whether or not the following (in C++ syntax) is legal (the comments above each line explains what that line does in case you are unfamiliar with C++ syntax):

```
/* Create a new object of class D and assign the reference to a variable d which
can hold pointers to objects of type D */
D *d = new D;
/* assign d to a variable 'a' which can hold pointers to objects of type A */
A *a = d;
```