COSC 2P90 MIDTERM TEST
OCTOBER 30TH, 2006

Time: 3:30 p.m. – 4:30 p.m. Total marks: 20 Total pages: 2
This is a *closed-book* exam No calculators are permitted
Answer all questions *in your exam booklet*.

Attempt question **4** and any **two** of questions **1** through **3**. If you answer all of questions 1 through 3 just the first two answers in your booklet will be marked. Be sure to answer all parts of the questions you choose. Questions 1 through 3 are worth a total of 6 marks each, and question 4 is worth a total of 8 marks.

1. What are the three alternatives for binding times in programming languages(1 mark)? Differentiate between static and dynamic scope in terms of bindings (i.e., what kind of binding is involved?) and binding times(2 marks). Differentiate between statically-typed and dynamically-typed languages in terms of bindings (i.e., what kind of binding is involved?) and binding times(2 marks). What is meant by declaration-reference binding(1 mark)?

2. Consider the following loop written in Pascal-like syntax. Explain 2 different possibilities for the calculation of the final value and increment of the loop index `i` (2 marks). What is the value of `j` after executing the loop in both alternatives (4 marks)?

```
j := 2;
for i := 1 to 2*j+1 step j+1 do
   j := j+1;
end;
```

3. Explain the concept of information hiding (2 marks). Why is this concept important in constructing large systems (2 mark)? Describe how packages in Ada support this idea (2 marks).

4. Explain each of the following parameter passing mechanisms (4x1 marks): call by value, call by value-result (not the Algol-W version), call by reference and call by name. Determine the value of the array `a` after executing the following program if the parameter is passed using each of call by value, call by value-result (not the Algol-W version), call by reference or call by name (4x1 mark):

```
program prog;

var a : array[1..2] of Integer;
var index : Integer;
```

```
procedure p(x : Integer);
begin
     index := 1;
     a[2] := 4;
     x := x + 2;
end;

begin
     a[1] := 0;
     a[2] := 0;
     index := 2;
     p(a[index]);
end.
```