Computer Science & Engineering 505 – Midterm

November 6, 1991

Closed Book – 50 minutes – 10 points per question – 80 points total

Name: _____

Please write any long answers on a separate piece of paper.

1. (a) Define the terms "statically typed" and "strongly typed". Are there any languages that are statically typed but not strongly typed? If so give an example. Are there any languages that are strongly typed byt not statically typed? If so give an example.

   (b) Consider the following Ada fragment.

```
subtype LITTLE is INTEGER range 0 .. 10;
J : LITTLE;
K : INTEGER;
K := 20;
J := K;
```

   Suppose that somebody made the following claim:

   > A program containing this fragment will compile and type-check correctly in Ada. However, it will give a runtime error, which in fact will indicate a type violation, since we are trying to assign an element (20) to a variable whose type specifies that it can only contain integers between 0 and 10. Since this type violation isn't detected until runtime, Ada isn't statically typed.

   Do you agree or disagree? Why?

2. Consider the following program in an Algol-like language.

```
begin
integer barnacle;
procedure clam(n: integer);
    begin
    n := n+8;
    barnacle := barnacle+3;
    print(n);
    print(barnacle);
    end;
barnacle := 0;
clam(barnacle);
print(barnacle);
end;
```

   What is the output if `n` is passed using:

   (a) call by value

   (b) call by value result

   (c) call by reference

3. What is the output from the following program in an Algol-like language?

```
begin

procedure pine(n: integer, p: procedure);
  begin
  procedure oak(k: integer);
    begin
    print("in procedure oak -- n="); print(n); print(" k="); print(k);
    end;
  if n<10 then
      begin
      if n=3 then pine(n+1,oak) else pine(n+1,p)
      end;
  if n=10 then p(n);
  end;

procedure nettle(k: integer);
  begin
  print("in procedure nettle");
  end;

pine(1,nettle);
end;
```

4. Suppose that the following Miranda script has been filed in.

```
add5 x = x+5
cube x = x*x*x
twice f x = f (f x)
mymap f [] = []
mymap f (a:x) = f a : mymap f x
double [] = []
double (a:x) = a : a : (double x)
second x y = y
mymember [] x = False
mymember (a:s) x = True, if a=x
                 = mymember s x, otherwise
```

What is the result of evaluating the following Miranda expressions? (If there is a compile-time type error, or a run-time error, or a non-terminating computation, say so.)

(a) `twice add5 3`

(b) `map cube [1..]`

(c) `twice double [1..5]`

(d) `mymember [1..]  0`

(e) `second (mymember [1..]  0) 3`

5. For the same Miranda script given in Question 4, what is the type of

   (a) `twice add5`

   (b) `mymap`

   (c) `mymap double`

   (d) `second`

   (e) `mymember`

6. Suppose we have an Ada program with two tasks `T1` and `T2`. `T1` includes an entry `ZORK`, and `T2` invokes `T1.ZORK`.

   (a) When can `T1` and `T2` run concurrently, and when is one or the other of them blocked?

   (b) To make this more concrete, here is an Ada program with two such tasks. What are the different possible outputs from this program?

```
with TEXT_IO; use TEXT_IO;

procedure DEMO is

task T1 is
   entry ZORK:
   end T1;

task body T1 is
begin
   loop
        select
           accept ZORK do
             PUT("in accept in T1"); NEW_LINE;
             end ZORK;
           PUT("outside of accept in T1"); NEW_LINE;
         or terminate;
      end select;
    end loop;
end T1;

task T2 is
end T2;

task body T2 is
begin
PUT("before invoking ZORK"); NEW_LINE;
T1.ZORK;
PUT("after invoking ZORK"); NEW_LINE;
end T2;

begin

end DEMO;
```

7. Both Fortran, and Pascal as defined by Jensen and Wirth, are weakly typed. Why?

8. (a) What is the difference between call-by-name in Algol 60 and lazy evaluation in Miranda?

   (b) What would be the effect on the semantics and efficiency of Miranda if it used call-by-name rather than lazy evaluation?

   (c) What would be the effect on semantics and efficiency of Algol 60 if it used lazy evaluation rather than call-by-name?