Computer Science & Engineering 505 – Midterm

November 6, 1992

Closed Book – 50 minutes – 10 points per question (except Question 7, which is 15 points)

75 points total

Name: _____

Please write any long answers on the back of the exam or on a separate piece of paper.

1. What does it mean for a program to be type correct? What does it mean for a program to be statically type correct? What is the difference? Name a language (if one exists) in which:

   (a) every program that compiles without errors is both type correct and statically type correct

   (b) every program that compiles without errors is type correct, but not necessarily statically type correct

   (c) every program that compiles without errors is statically type correct, but not necessarily type correct

   (d) some programs that compile without errors are not statically type correct

   (e) some programs that compile without errors are not type correct

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

```
begin
integer barnacle;
procedure clam(j,k: integer);
    begin
    print(j,k);
    barnacle := barnacle+3;
    print(j,k);
    end;
barnacle := 1;
clam(barnacle, barnacle+barnacle);
end;
```

   What is the output if `j` and `k` are both passed by:

   (a) call by value

   (b) call by name

   (c) call by reference

3. Compare how multiple inheritance is handled in Extended Smalltalk and CLOS.

4. 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<5 then pine(n+1,oak);
  p(n);
  oak(n);
  end;

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

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

5. Consider the following class hierarchy in Smalltalk:

```
Object
   Politician
      Republican
      Democrat
```

(In other words, `Politician` is a subclass of `Object`. `Republican` and `Democrat` are subclasses of `Politician`.) Suppose the following methods are defined:

```
Politician
   pressConference
       ^ 'If elected, I promise that there will be ' ,  self soundBite
   soundBite
       ^ 'more apple pie'
   TVad
       ^ self soundBite , ' and ' , self soundBite

Democrat
   soundBite
       ^ 'more jobs'

Republican
   soundBite
       ^ 'no new taxes' , ' and ' , super soundBite
```

If we have instances of `Politician`, `Democrat`, and `Republican`, and send each of them the message `pressConference`, what will happen? Explain how the appropriate method(s) are found. Note that strings understand the message "," (comma) to mean concatenation.

6. Suppose one re-implemented the political example in Question 5 in a prototype-based object-oriented language, but otherwise following the original Smalltalk code as closely as possible.

   (a) Suppose delegation were being used. Explain what happens when a Democrat is sent the `pressConference` message.

   (b) Suppose that one tried to handle this situation by having a Democrat object, with a Politician object as one of its instance variables. Suppose rather than using delegation, when the Democrat received the `pressConference` message, it just sent it to the Politician object. What would happen?

7. True or false?

   (a) In Smalltalk-80, while loops are defined using the `whileTrue:` message, which is sent to a block.

   (b) In Smalltalk-80, every object is an instance of some class. Classes are themselves objects, and each class is an instance of a different metaclass. Each metaclass is an instance of a different meta-metaclass, and so forth. However, to stop the infinite regress, the system only stores 100 levels of meta-meta-meta ...metaclasses, and creates the remaining levels on demand.

   (c) A "thunk" is an implementation technique for passing parameters by value in Algol-60.

   (d) Blocks in Smalltalk-80, like procedures in Algol-60, are not first-class citizens. They can only be passed as parameters, but can't be returned from a method or assigned to a global variable, since if this were done the Smalltalk implementor wouldn't be able to allocate activation records on a stack.

   (e) A FORTRAN loop `DO 100 I=1,N` will execute zero times if N=0.

   (f) Recursive subroutines aren't supported in standard FORTRAN-IV.

   (g) It is possible for users to define their own kinds of iterators in Smalltalk-80, for example, to traverse a data structure in a given order.

   (h) In Ada, a handler for an exception can optionally retry execution of the code that raised the exception.

   (i) In Ada, if the handler for the `ZORK` exception re-raised the `ZORK` exception, the program would get into an infinite loop, since the same handler would get invoked again for `ZORK`, and so forth.

   (j) It would be possible to define the basic control structures (conditionals and loops) in Smalltalk — they don't need to be primitives.

   (k) COMMON blocks in FORTRAN allow data to be shared among subroutines and the main program, without passing the information as parameters to the subroutines.

   (l) Double-dispatching is a technique for simulating multiple dispatch in a language such as Smalltalk that only supports single dispatch.

   (m) When executing a rendezvous in Ada, the calling task suspends while the called tasks executes the sequence of statements in the accept statement. Thereafter, the two tasks can continue their execution in parallel.

   (n) Ada allows parameters to be passed using both positional and keyword notation; however, in a given call, the notations can't be mixed.

   (o) True/false questions have no place on an examination in a graduate course; the only time they are given is when instructors are too lazy to grade penetrating page-long answers to essay questions.