symbol tables were on the first exam of this particular year's exam. We did not discuss register allocation in this particular year.

This exam has questions from previous CISC 471/672.
Not all questions are applicable to this semester, as the project and some topics differ between semesters. This exam was during a semester that built a compiler from Decaf language to TAC intermediate code.

I . General questions (10 points)
1. (1 point each) True/False questions: Please answer true (T) or false(F) for the statements below. You may provide a short explanation. Marks that are unclear will be counted as wrong.

_______1. Decaf is statically typed and strongly typed.

_______2. Reachability in garbage collection is just an approximation of garbage.

_______3. All syntax checking can be done by the parser.

_______4. The memory space for programs, including code and runtime structures, need not be contiguous.

_______5. Access (static) links point to the dynamic parent of a procedure/method.

_______6. A display is used to speed up the traversing of the call (control chain).

_______7. A symbol table of a method/procedure is always deleted once the code is generated for that method/procedure.

_______8. In a strong statically typed language, once semantic analysis is done, there are no more syntactical or type errors that can be caught.

_______9. When the length of an array is known at compile time, storage for the array can be placed in the activation record.

_______10. Objects referenced by pointers are stored in the heap.

II. Short answers

1. (5 points) Describe the tradeoffs between (1) static type checking, (2) dynamic type checking and no type checking. 1

1. **(** 9 points) Explain how the inferred type of an overloaded function is resolved during type checking. Contract this with the handling of overridden methods. For extra credit (5 points), define and show examples of each of these. 1

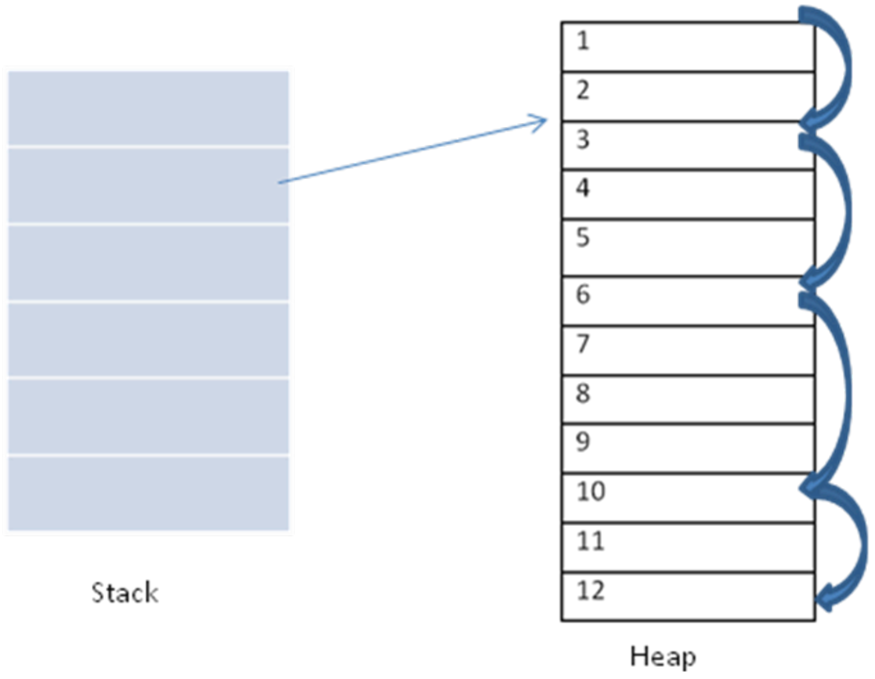
3. (10 points) Describe the actions of the type checking/inferencer when implementing the following type rule: 1

$$\frac{\begin{array}{c} f \text{ is an identifier.} \\ f \text{ is a non-member function in scope S.} \\ f \text{ has type } (T\ , \ldots, T\ ) \rightarrow U \\ S \vdash e_i : R_{i \text{ for } 1 \leq i \leq n} \end{array}}{S \vdash f(e_1\ , \ldots, e_n\ ) : U}$$

III Problems

8

1. (10 pts) Garbage Collection:. Show the memory, including any extra counters or space, when (1) mark and sweep and (2) stop and copy when they finish collecting garbage.



2. (10 points) For the expression, $a = b + c * d – 5 + c * d$, show (a) a concrete syntax tree (assuming typical grammar for expressions), (b) abstract syntax tree, and (c) three-address code representations. Assume that * is higher precedence than + and – which are the same precedence. All operators are left associative.

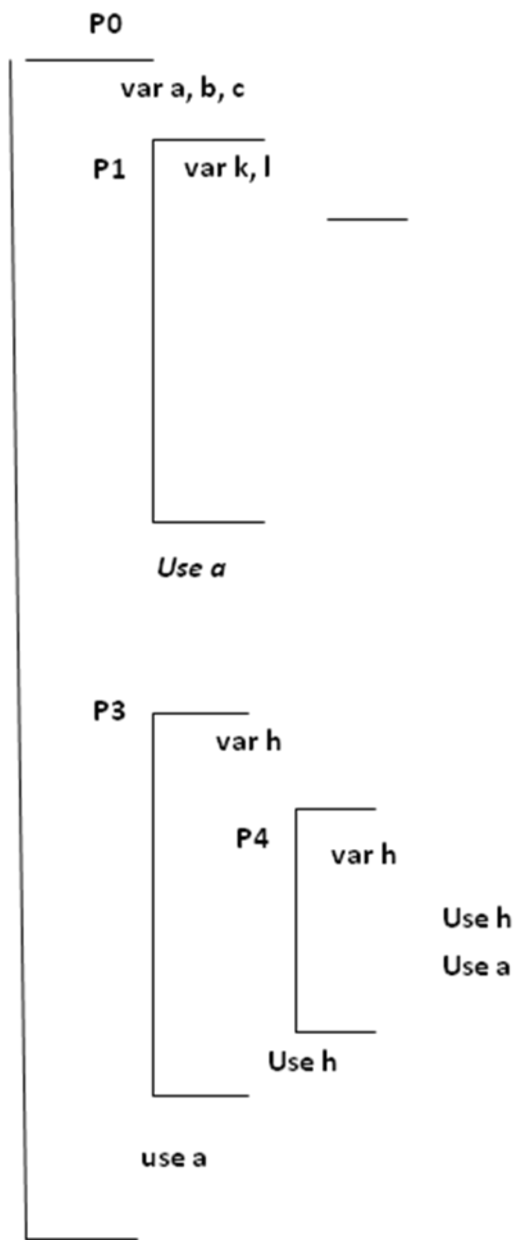
3. (12 points) Consider the following class hierarchy:

```
Class A                 Class B extends A            Class C extends B
   Int x, y;                      int x, t, s;                      int t, y, q;
   Method v;                      Method u;                         Method n;
   Method n;                      Method n;                         Method u;
                                                                    Method m;
End A                       End B                       End C;
```

a. (6 points) Draw the layout of an object for each of the classes above.

b. (6 points) Draw the vtables for each of the classes above.

4. (12 points) Given the following program skeleton,
   a. show the symbol table and stack, including all ARs, when P0 calls P1 calls P3 calls P4 calls P1. Give the references for the variables that are used.
   b. Assume P1 and P4 terminate. Show the stack then.



5. (12 points) Consider the following programming language statement in an object-oriented language in which this is syntax for a method call.

6

```
u.G(p,q);
```

a. (6 points) Give a TAC segment that implements this call, given that u points to an object of class C whose first method is G, and C inherits one method F from its parent class B.

b. (6 points) Describe the steps in pseudocode or TAC code for implementing a new object allocation, for instance, a new Cow that is referenced by Bob, through the instruction Bob = new(Cow).

6. (12 points) For each of the following language features, indicate the simplest allocation scheme that will suffice by labeling the feature with either static, stack, or heap. Note all features could be allocated at runtime using the heap, but the simplest allocation mechanism is preferred for runtime savings. 8

(1) _______objects referenced by pointers
(2) _______array of fixed length declared local to a function
(3) _______integer type variable declared in the global declaration section of a program
(4) _______activation records for recursive functions
(5) _______array of unknown length until function entry and then fixed throughout the current activation
(6) _______varying length strings with concatenation operations
(7) _______objects in object-oriented languages allocated with new
(8) ______ call by value parameters of fixed size

Have a nice relaxing winter break!