CORE EXAMINATION
Department of Computer Science
New York University
May 15, 1998

This is the common examination for the M.S. program in CS. It covers core computer science topics: Languages and Compilers, Operating Systems, and Algorithms. The exam has two parts. The first part lasts three hours and covers the first two topics. The second part, given this afternoon, lasts one and one-half hours, and covers algorithms.

Attempt all of the questions. You are not required to take the algorithms section of the exam if you have passed the FOCS exam in the past.

Use a separate booklet for each question. Mark each booklet with your exam number and with the question it contains. Do not answer more than one question per booklet.

You will be graded according to your exam number, shown on the envelope containing the booklets. Remember your exam number: when grades are given out, they will be published according to this number, not by name.

Make sure your name and signature are on the envelope. This is the only place where your name appears. Please include all the booklets inside the envelope. You can keep the exam.

Good luck!

Question 1

a. In a few sentences, explain how inheritance contributes to data abstraction and code reuse.

b. What does the following fragment print?

```
class A {
 public: void mf () {cout << "I am an A." << endl;
}

class B : public A {
   public: void mf () { cout << "I am a B" << endl;
}

B x;
A *pB = &x;
B *pA = &x;
pB -> mf ();
pA -> mf ();
```

c. Does the answer to (b) change if method mf is declared virtual? explain.

Question 2

next booklet please.
Many programming languages have constructs that manipulate slices, that is to say contiguous portions of one-dimensional arrays. For example, in Ada slices can be used wherever array values can be used, which includes expressions, assignments, and actuals in subprogram calls:

```
A (x .. y) := B (s .. t);                          (1)
if  s (1..3) = (11, 19, 37) then ...               (2)
Modify (A (x+1 .. y-1));                           (3)
```

a. Write a grammar for expressions that includes slices. You can use the syntax of Ada, or that of any other language you know that supports slices. Make sure that slices of slices are legal. Your grammar should include arithmetic operators and indexed expressions, that is to say array components.

b. Describe briefly the semantic checks that have to be performed to verify that a slice expression is correct.

c. Assuming that the bounds are general expressions, describe the run-time checks that must be performed for the assignment (1) to be legal.

d. Using quadruples, or the assembly language of your choice, describe the code that has to be generated for the assignment (1) . Assume that the arrays A and B are known to be disjoint, and the bounds are variables.

e. Most programming languages do not support slices of multidimensional arrays. Why are multidimensional slices more complicated to implement? (Hint: think of slicing a two-dimensional array along its second dimension).

Question 3

next booklet please.

a. Explain the terms error detection and error correction as they apply to compiler construction. Give simple examples of each.

b. Explain the notion of a warning message. Give a typical example. Why is there a distinction between warning messages and error messages?

c. In a language like C or Ada, it is an error to reference a variable before it is initialized . A compiler can sometimes detect this error at compile time, but not always. Give examples of cases where the errror can and cannot be detected at compile time.

d. Compilers often distinguish between syntax errors (discovered by the parser), and semantic errors (discovered by the analyzer). But sometimes errors that are definitely syntactic errors according to the official language grammar are treated as semantic errors. Why? Give an example.

e. The layout of the program, i.e. use of whitespace, is irrelevant to the meaning of a program in C or Ada, but is sometimes useful for determining the best possible error message. Give an example where the program layout suggests the error message most likely to help the user.

Question 1

next booklet please.

a. Nearly all modern systems with demand paging use pages of about the same size (8KB give or take one or possibly two powers of 2). Since these systems do NOT use huge pages (say > 128KB) or tiny pages (say < 1KB) there must be problems with very large pages and with very small pages. DESCRIBE these problems.

b. Imagine a technology change in paging disks so that both seek time and rotational latency take zero time (i.e. the only delay is the time to transfer the requested page). Would this change argue for larger or smaller pages or would it not effect the choice of page size? JUSTIFY your answer.

c. Now imagine that disks remain as they are today but memory prices decrease 100 fold so that personal computers have over a gigabyte (1000MB) of RAM. Would this change argue for larger or smaller pages or would it not effect the choice of page size? JUSTIFY your answer.

Question 2

next booklet please.

Consider the following preemptive priority-scheduling algorithm based on dynamic priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (on the ready queue, but not running) its priority changes at rate a; when it is running, its priority changes at the rate b. All processes are given a priority of 0 when they enter the ready queue. The parameters a and b can be set to give many different scheduling algorithms.

a. What is the algorithm that results from $b > a > 0$? Explain.

b. How can this scheme be modified (choices for a, b, and initial values) to obtain round-robin scheduling?

CORE EXAMINATION
Department of Computer Science
New York University
May 15, 1998

This is the common examination for the M.S. program in CS. It covers core computer science topics: Languages and Compilers, Operating Systems, and Algorithms. The exam has two parts. The first part lasts three hours and covers the first two topics. The second part, given this afternoon, lasts one and one-half hours, and covers algorithms.

Attempt all of the questions. You are not required to take the algorithms section of the exam if you have passed the FOCS exam in the past.

Use a separate booklet for each question. Mark each booklet with your exam number and with the question it contains. **Do not answer more than one question per booklet**.

You will be graded according to your exam number, shown on the envelope containing the booklets. Remember your exam number: when grades are given out, they will be published according to this number, not by name.

Make sure your name and signature are on the envelope. This is the only place where your name appears. Please include all the booklets inside the envelope. You can keep the exam.

Good luck!

Basic Algorithms

Question 1 There is a natural correspondence between binary trees and (legal) parentheses expressions. Given a binary tree T, define its parentheses expression P_T as follows. If T is the empty tree, then P_T is the empty string. Otherwise, let L and R be the left and right subtrees, respectively, of (the root of) T; then $P_T = (P_L)P_R$. For example, if T is the complete binary tree with 2 levels (and hence 3 nodes), then $P_T = (())()$.

a. (5 points) Describe an efficient algorithm that, given a binary tree T, computes its parentheses expression P_T. What is the running time of your algorithm in terms of $|T|$, the number of nodes in T?

b. (5 points) Describe an efficient algorithm that, given a legal parentheses expression s, computes the (unique) binary tree T such that $P_T = s$. What is the running time of your algorithm in terms of $|s|$, the number of parentheses in s?

Question 2 (next booklet, please)
There are N computers connected in a large but haphazard network. Each computer has a list of other computers that it is connected to. A message can get from any machine in the network to any other machine in the network, but only through a sequence of "hops" in which a message is transmitted from a machine to another that it is directly connected to. If machine x is directly connected to machine y, there is a delay, $d(x, y)$, which is the time it takes to transmit the message directly from x to y. Assume that the delay caused by the machines themselves is insignificant compared to the transmission delays. We want to transmit a message from a source machine, s, to a target machine, t.

a. (5 points) Describe an algorithm that determines a path from s to t that minimizes the total delay. The total delay is the sum of the delays for all the hops in the path. The work for your algorithm should be $O((N+C)\log N)$ where C is the number of direct connections between computers.

b. (5 points) Now we want routes that do not exceed a given "hop count", k. The hop count of a route is the total number of hops in the route. Find an algorithm related to dynamic programming that determines the minimum delay among paths with k or fewer hops from s to u, for all $k < k_{\max}$ and all u (but s fixed). Your algorithm should use an array, $md[k][u]$, that stores this minimum delay information. The values $md[0][u]$ are easy to compute. If all the values $md[k-1][u]$ have been computed, we can find the $md[k][u]$ by considering the last hop in the minimum delay path with k hops. What is the running time of your algorithm?

Question 3 (next booklet, please)

Suppose you are given the problem of finding in sorted order the k smallest integers in an array of size n, where k is much smaller than n but much larger than 1.

a. (5 points) Describe how selection sort, mergesort, heapsort, and quicksort can be adapted to this problem. (There is nothing to be done with insertion sort). Your description need not give the pseudo-code for the modified algorithms; it is enough to describe clearly what changes can be made.

b. (3 points) Find the worst-case running time of the modified selection sort.

c. (2 points) Show that any comparison-based method for solving this problem must take at least time $\Omega(k \log n)$ in the worst case.