APPLIED MATH 101-102

An Introductory Blurb
-or-
It's Too Late Now, Baby

September 19, 1972
TABLE OF CONTENTS

1 PHILOSOPHY 1

2 THE ANIMALS IN THIS ZOO 2

3 THE OPEN-DOOR POLICY 3
  3.1 Questions and Answers 3
  3.2 Andries van Dam (AvD/Andy) 3
  3.3 Robert g. Munck (RgM) 4
  3.4 Ken Magel 4

4 ARRANGEMENTS 5
  4.1 Times and Places 5
  4.2 Class Attendance 5
  4.3 Accounts 5

5 COMPUTING LABORATORY 6

6 OTHER STUFF 7
  6.1 Collaboration 7
  6.2 Documentation 7
  6.3 Lateness 7
  6.4 Manual Fees 7
  6.5 Special Goodies 7

7 SCHEDULE 8

8 ASSIGNMENTS 9
A.M. 101-102 is called Foundations of Computer Science. The purpose of this course is to dispel the notion of computing as a "black box" activity (or even a pastel blue one) and to remove each and every mystery whenever possible. As such, it will attempt to investigate, on a fundamental level, a good number of the areas in (non-numerical) Computer Science, including machine architecture, programming languages, list processing, formal language theory, and compiler design. By exposing you to a broad range of topics, the course can serve as a good basis for the continued study of the science, both here (in courses such as 193b, 273-274, 277-278...) and at another graduate school, or for a practical position in industry, otherwise known as the "real world" (FEH!).

This year, the course is intended as a sequel to A.M. 44. Thus, it assumes a prior knowledge of some machine/assembly language and of basic PL/I. In addition, because of the broad range of topics and concentrated study, it assumes that you are interested in more than just learning to code programs. As a matter of fact, it is hoped that A.M. 101-102 will cure the "coding jock" syndrome. A coding jock is a creature who spends all night coding and debugging and then sleeps through a theory class. In other words, to enjoy the course you must be a bit of a computer scientist at heart.
2 THE ANIMALS IN THIS ZOO

Andries van Dam, Associate Professor of Computer Science, recently returned from a year's hiding in a small foreign country, is ready to teach everything he knows concerning logical design and language theory. Room 305, Applied Mathematics Building; Extension 3089.

Robert G. Munck, Systems Programmer - Computing Laboratory. Recently returned from a summer's hiding in his office at the Bowels of the Lab. Always ready to lecture on Assembly language, PL/I, and operating systems. Room B-6, Computing Laboratory; Extension 6.

A veritable horde of graders:

<table>
<thead>
<tr>
<th>NAME</th>
<th>PHONE(S)</th>
<th>SORE SPOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paul C. Anagnostopoulos &quot;Aggie&quot;</td>
<td></td>
<td>50-blip assemblies</td>
</tr>
<tr>
<td>Carol Chomsky &quot;Carol&quot;</td>
<td></td>
<td>grammar types</td>
</tr>
<tr>
<td>Kenneth I Magel &quot;Ken&quot;</td>
<td></td>
<td>large compilers</td>
</tr>
<tr>
<td>Robert Schomp &quot;Bob&quot;</td>
<td></td>
<td>trailing edits</td>
</tr>
<tr>
<td>John Woodward &quot;Woody&quot;</td>
<td></td>
<td>backup tapes</td>
</tr>
</tbody>
</table>

And finally, various and sundry hangers-on, phantoms from the past, general flamers, and a dog or two.
3 THE OPEN-DOOR POLICY

3.1 Questions and Answers

As listed, there are five graders for this course, provided for your consultation and entertainment. They should be able to answer (by now, we hope!) any legitimate questions and queries. They are not responsible, however, for solving the assignments or finding each and every one of your bugs - this is a job for you. Furthermore, before you approach a grader, you should try to think out the problem and answer it yourself.

To avoid a rash of muggings by making you walk between the lab and a consultant's office, a grader will be on duty in the Keypunch Operator's room every night of the week, including weekends. He will be equipped with a complete manual library and should be the first recipient of your questions, but if his or her answer is unsatisfactory, other graders can be sought out. Consultation will most likely be from 8:00 to 10:00 P.M.

In past years, there were also a class of human beings known as "lab bums". These people can be identified by the fact that they eat, sleep, and watch TV in the Lab even if they have nothing to do there. However, this year, because Computer Science is developing into a more serious, theoretically-oriented discipline here at Brown, these lab bums have migrated to Richardson Hall, where they engage in drinking beer, playing chess, and an occasional absurd meeting.

3.2 Andries van Dam (AvD)

This man is the official Professor for the course, and hence is the ultimate receptacle for all complaints and suggestions. In past years people have occasionally been reluctant to voice dissent concerning the course - this year we would like to see a change. If you have anything to say about course content, methods of presentation, speed, grading algorithms, etc., etc., feel free to talk to the graders and/or AvD at any time. In particular, if you want to dump on a grader, do so - we only have five of the fools and they'd better be worth the crummy salary we pay them.

AvD will have official office hours on Monday and Wednesday from 3:00 to 3:30. However, his door is always open - so come in any time. Always a good chance that a grader will be around there, too.
NOTE: Please rest assured that we bear no grudges and that any comments on your part will not affect our judgement in any way. If you believe that, we're amazed.

3.3 Robert G. Munck (RgM)

"If you have an intelligent question then you're probably smart enough to answer it yourself, and if you have a dumb one I don't want to hear it anyway, so don't ask me any questions." In spite of the way this rather harsh quote from RgM sounds, he's really a rather friendly and helpful fellow, once you get to know him. However, he's also busy as hell trying to keep the machine on which you clods are running in shape. If a grader can't answer your question, and he doesn't follow up on it within a reasonable period, and you've asked him two or three times, then it's legitimate to go to RgM for advice. Otherwise, try not to bug him. The table on his door tells you where he is at all times, except when he doesn't want you to know.

3.4 Ken Magel

Ken Magel is euphemistically the "Head Grader" for A.M. 101-102. In addition to specializing in running around with his head cut off (i.e., like a chicken), he is responsible for keeping the other graders in line (hence, he is bigger than any of the them). All complaints and, perish the thought, compliments, should eventually end up in Ken's hands, and hopefully any necessary corrective action will be taken.

MANUAL FEE - $10 - Bob Schamp
4 ARRANGEMENTS

4.1 Times and Places

4:39, Chicago, 9:27, 8:16, Cranston, East Timbuctu,
Greenwich Village, 12:59.

Furthermore, you might be interested in knowing when and
where the course is going to meet. We propose to have
three regular hour-and-a-half classes on Monday and
Wednesday at 3:30 and Tuesday at 11:30. Furthermore,
there will be a sometimes mandatory/sometimes optional
fourth class on Friday at 3:30. The 3:30 classes will be
held in Barus and Holley 166, except when occasionally
pre-empted by some Engineering conference or another, and
the 11:30 class in Whitehall 301 (sorry).

You can be fairly confident that classes will end
promptly after an hour and a half. If a class begins to
go over, simply whisper across the aisle to a grader, who
will proceed to erase the blackboard or sweep the floor.
This will act as a signal to all those present, who can
then just get up, YAWN, and leave.

4.2 Class Attendance

You are responsible for all material covered and
announcements made (both oral and written) during each
class, whether or not you actually attend. A schedule is
included in this blurb, which will be kept up-to-date
with additions and/or corrections. Any changes in
schedule, assignments, protocol, etc., will be announced
in class and handed out in a blurb.

4.3 Accounts

Each of you will be given an account number, good for
interactive use or for batch (what's batch?). You are
expected to do all of your programming assignments using
this number, and a limited amount of your own work is
permitted. Please keep in mind, however, that A.M.
101-102 is not sponsored by Dean Hazeltine; we do not
have unlimited funds. Initially your account will
contain $150 - when it begins to run low, see Ken Magel
and he will fill it back up. Try to conserve runs and
debug more than one thing at a time; you are limited to
a total of $600 for the semester.
The Brown University Computing Laboratory is run as an independent service facility, reporting to the Provost, - not by the Division of Applied Mathematics or the Center for Computer and Information Sciences. Its facilities are for the use of all departments in the university. Students in A.N. 101-102 must realize that they are but a small segment of the lab users and only have rights equal to those of other users. They should neither ask for nor expect any special privileges.

In general, people in the lab fit into one of three categories. The first is that of the USER. A user is a person who has somehow gotten the impression that the computer can be used as a tool, and is not in computing for its own sake. Users should not be bothered by loud voices or boisterous conduct or even questions. Furthermore, you should not develop a "more righteous than thou" attitude toward them; you're a lousy student and they're doing important research. Yessir. The second group of people is the LAB STAFF, who have recently dwindled down to a mere handful for reasons both you and I don't want to know. This group is basically friendly toward you, but look at them the wrong way and they’ll become very annoyed. A smile and a please/thank you go a long way; keep them on our side. The final group is the COMPUTER SCIENTISTS themselves. These are your friends and are generally willing, at least up to a point, to lend a hand and answer a question. You better not mind them too much because you yourself may be one some day.

Our people can usually be found in the BUGS room or among the keypunchers. A good laugh can be had between runs by standing outside the BUGS room and listening to the members of the graphics project talking their language (...microp... ...control store... ...XMALE... ...joystick... ...Y and Z...). It is something like a mixture of Greek, Sanskrit, and nonsense. These people, along with the rest of the computer scientists, are really human once you get to know them.

Two important points: do not bug the dispatcher about when your run will be back unless an unreasonable amount of time has passed; "unreasonable" meaning a long time. Also, clean up after yourself - all listings and cards and papers should be discarded or removed. Making a nuisance out of yourself(ves) is likely to get us thrown out of the lab.
6 OTHER STUFF

6.1 Collaboration

Due to the nature of your work in A.M. 101-102, with students constantly seeing one another in and around the lab, it is inevitable that consultation between people should occur. Indeed, this type of collaboration is encouraged to promote a "project-people"-type atmosphere. However, undue collaboration and copying of work is not condoned. We realize the obscurity of this statement—if you are at all unsure of discussing some topic with a fellow student, come discuss it with a grader.

6.2 Documentation

As in A.M. 44, we expect detailed documentation with each program, including a fairly macro flowchart, a well-written ELD (maybe you know English?), and a run history. In addition, copious comments should be present in the source code itself. Remember that the graders really enjoy humor in what you hand in—stuff that is too serious makes them unhappy, and when they become unhappy, ...

6.3 Lateness

In A.M. 101-102, unlike 44, we are expecting to get assignments turned in on time, and lateness will be penalized. The penalty will be a function of your sob story (which should be included in the run history), how well the assignment was completed, and how late you actually do get it in.

6.4 Manual Fees

There will be a $10 fee charged to all students covering the cost of XEROXing notes, assignments, announcements, etc. This money should be paid to Bob Schomp as soon as possible.

6.5 Special Goodies

As you get on through the course, you may find yourself becoming especially interested in one aspect of Computer Science. It is highly probable that one of the projects in which AvD and his cohorts are engaged is related to this interest. If this should happen, just see the nearest grader and you will be directed to the appropriate person. Furthermore, fame and wealth may come your way in the form of a grader job next year (if there is one).
NOTE that this course may become habit forming. Loss of nights' sleep, weekends, and/or vacations may occur. All the graders have ulcers anyway, so what the hell...
The following schedule is hopefully an accurate reflection of what we intend to teach and when we intend to teach it. Corrections and updates will be handed out whenever necessary to keep it up-to-date. Lectures marked with an asterisk will probably be optional.

<table>
<thead>
<tr>
<th>DATE</th>
<th>WHO</th>
<th>LECTURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept. 19</td>
<td>Organization</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Overview of the course</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>RgM</td>
<td>System/360: Base-Displacement addressing</td>
</tr>
<tr>
<td>25</td>
<td>RgM</td>
<td>PSW; States of the machine; Alignment</td>
</tr>
<tr>
<td>26</td>
<td>RgM</td>
<td>Condition codes; Branching; Interrupts</td>
</tr>
<tr>
<td>27</td>
<td>AvD</td>
<td>Boolean Algebra</td>
</tr>
<tr>
<td>29</td>
<td>AvD</td>
<td>Propositional Calculus</td>
</tr>
<tr>
<td>Oct. 2</td>
<td>RgM</td>
<td>Debugging assembly language</td>
</tr>
<tr>
<td>3</td>
<td>AvD</td>
<td>CMF/DMF</td>
</tr>
<tr>
<td>4</td>
<td>AvD</td>
<td>Combinatorial Circuits</td>
</tr>
<tr>
<td>6</td>
<td>RgM</td>
<td>CSECTs, DSECTs, and linking</td>
</tr>
<tr>
<td>9</td>
<td>RgM</td>
<td>Assembly language wrap-up</td>
</tr>
<tr>
<td>10</td>
<td>AvD</td>
<td>Minimization; Karnaugh Maps</td>
</tr>
<tr>
<td>11</td>
<td>AvD</td>
<td>Quine-McCluskey minimizations</td>
</tr>
<tr>
<td>13*</td>
<td>AvD</td>
<td>Don't care optimization</td>
</tr>
<tr>
<td>16</td>
<td>RgM</td>
<td>PL/I based variables</td>
</tr>
<tr>
<td>17</td>
<td>AvD</td>
<td>Logical operations; Simple adder</td>
</tr>
<tr>
<td>18</td>
<td>AvD</td>
<td>Encoders/Decoders</td>
</tr>
<tr>
<td>20*</td>
<td>AvD</td>
<td>Basic components and gate realization</td>
</tr>
<tr>
<td>23</td>
<td>AvD</td>
<td>Sequential circuits</td>
</tr>
<tr>
<td>24</td>
<td>AvD</td>
<td>FSMs; Sequential machines; state tables</td>
</tr>
<tr>
<td>25</td>
<td>RgM</td>
<td>Debugging PL/I</td>
</tr>
<tr>
<td>27</td>
<td>RgM</td>
<td>PL/I implementation and environments</td>
</tr>
<tr>
<td>30</td>
<td>AvD</td>
<td>Flow diagrams and FSMs</td>
</tr>
<tr>
<td>31</td>
<td>RgM</td>
<td>Stream I/O in PL/I</td>
</tr>
<tr>
<td>Nov. 1</td>
<td>AvD</td>
<td>FSA → flow diagram; Flow diagram → state table</td>
</tr>
<tr>
<td>3*</td>
<td>AvD</td>
<td>State table → flow diagram; State table → FSA</td>
</tr>
<tr>
<td>6</td>
<td>RgM</td>
<td>Record I/O</td>
</tr>
<tr>
<td>7</td>
<td>RgM</td>
<td>Record I/O</td>
</tr>
<tr>
<td>8</td>
<td>AvD</td>
<td>Clock pulses; Types of flip-flops</td>
</tr>
<tr>
<td>10</td>
<td>AvD</td>
<td>Shift registers; Counters</td>
</tr>
<tr>
<td>13</td>
<td>AvD</td>
<td>Primary storage</td>
</tr>
<tr>
<td>14</td>
<td>AvD</td>
<td>Simple processor block diagram</td>
</tr>
<tr>
<td>15</td>
<td>AvD</td>
<td>Instruction cycle decoding</td>
</tr>
<tr>
<td>17</td>
<td>RgM</td>
<td>PL/I pre-processor; Defined variables;</td>
</tr>
<tr>
<td>20</td>
<td>RgM</td>
<td>More PL/I</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>Microprogramming</td>
</tr>
<tr>
<td>27</td>
<td>RgM</td>
<td>Still more PL/I</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>Microprogramming</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>Microprogramming</td>
</tr>
</tbody>
</table>
8 ASSIGNMENTS

The following is a list of the assignments currently planned for 101. We are not adamant about these particular assignments and an extensive search of van Dam's contracts has failed to uncover any legal requirements. Nevertheless, these are approximately the assignments that have been given previously (TRADITION!!!).

1. Machine language program - 4 days (12 statements)
2. Assembly language program - 4 days (25 statements)
3. Assembly language program - 8 days (50 statements)
4. PL/I program - 5 days (48 statements)
5. PL/I program - 14 days (228 statements)
6. PL/I with Assembly language - 25 days (500 statements)
7. APL program - 4 days (3 statements)
8. APL program - 4 days (12 statements)
9. APL program - 6 days (30 statements)
10. APL program - 14 days (80 statements)

Assignments will generally overlap, so it is suggested that you begin work on an assignment immediately after it is given. You'll notice that it is required to learn APL for some of the assignments, namely the ones that pertain to the logical design part of the course. More will be said about this later.
"I took 44, that's how come."
I Wonder if he's auditing it or taking it for credit...