An Introductory Missive
September 16, 1974
Andy Malis
Dave Taffs
Jens Bll
Brad Silverberg
# Table of Contents

- Philosophy ........................................................................................................... 1
- The Inhabitants of this Asylum ........................................................................ 2
- The Open Door Policy ...................................................................................... 4
- Questions and Answers ..................................................................................... 4
- Criticisms and Comments ............................................................................... 4
- Consulting ........................................................................................................... 5
- Arrangements ..................................................................................................... 6
- Times and Places ............................................................................................... 6
- Class Attendance ............................................................................................... 6
- Accounts ............................................................................................................. 6
- The Computing Laboratory ............................................................................. 8
- Other Things ...................................................................................................... 9
- Collaboration ..................................................................................................... 9
- Documentation .................................................................................................. 9
- Lateness ............................................................................................................. 9
- Manual Fees ..................................................................................................... 10
- Special Goodies ............................................................................................... 10
- Schedule ........................................................................................................... 11
- Order of Assembler Topics ............................................................................. 11
- Order of Logical Design Topics .................................................................... 12
- Order of PL/I Topics ....................................................................................... 12
- Assignments ..................................................................................................... 14
- Medical Disclaimer ......................................................................................... 15
PHILOSOPHY

A.M. 101 is called Introduction to Software and Machine Architecture. The purpose of this course is to dispel the notion of computers as mystifying "black boxes" (or even IBM pastel blue ones) and to remove each and every mystery wherever possible. As such, it will attempt to investigate, on a fundamental level, a good number of areas in Computer Science, including logical design, machine architecture, programming in 5/360 Assembler and high level languages; and "microprogramming"; also list processing and information retrieval. By exposing you to a broad range of topics, the course can serve as a good basis for the continued study of computer science, both here and at graduate school, or for a practical position in industry, otherwise known around here as the "real world" (PEH?).

This year the course is designed as a sequel to A.M. 51-100, so it assumes a prior knowledge of some structured design and programming in machine/assembly language and of basic PL/I. In addition, because of the broad range of topics and concentrated study, it is assumed that you are interested in more than just becoming a "code jock", a strange creature that spends all night coding and debugging programs and then sleeps through a theory class. In other words, to enjoy this course you must be a bit of a computer scientist at heart.
THE INHABITANTS OF THIS ASYLUM

Approximately 30.991 "students".

Andries van Dam (AVD), Professor of Computer Science in the Division of Applied Mathematics. He is ready to teach all that he knows, and some that he doesn't, concentrating on logical design and machine architecture. He can usually be found in room 305, Applied Math building (otherwise known as the Evil Castle), extension 3088. Despite external appearances, he is basically friendly, and likes to be called Andy.

A virtual plethora of potential guest lecturers:

Craig Mathias (CJM) - Craig is known for having some of the most colorful wall decorations in the area. Now if some of that color would only sneak into his lectures...

Ken Magel (KIM) - Ken, a specialist when it comes to confusing programming languages, will lecture on PL/I and related matters.

Hal Webber (HHW) - Hard-working Hal, the only person around here with something to show for his work, may lecture on some aspects of logical design and the modern (?) aspects of computer engineering.

A veritable pride of graders:

Smilin' Andy Malis (AGM), "Head" Grader (whatever that means) - Andy has the great gift of being able to simply explain any subject in 20,000 words or more, and making it totally incomprehensible.

Uncle Dave Taffs (DAT) - Dave is one of the world's foremost experts on optimization, especially of burger runs, but he never lets it go to his head - only to his stomach.

---

1As you may know, computers sometimes have precision problems.

2As this is being written, we are not quite sure who the guest lecturers will be.
Friendly Brad Silverberg (BAS) - Brad believes in service with a smile, which suits him better for pumping gas than for grading this course.

Yumping Jens Dill (JMD) - Veteran of AM 100 grading and a wizard with assemblers, Jens will help us assemble this course into something coherent (maybe).

And finally, various and sundry hangers-on, phantoms from the past, general flamers, a dog or three, and anyone else confused enough to wander in.
THE OPEN DOOR POLICY

Questions and Answers

As listed, there are four graders for this course, provided for your consultation and entertainment. They should be able to answer any legitimate questions and queries. They are not responsible, however, for solving the assignments for you 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.

Generally, you should first direct any queries to the grader who is on consulting duty (see below). If the consulting grader is unable to help you, feel free to ask any of the other graders. In the unlikely case that they cannot help you, you can go to the lecturers of the course for help. Ultimately, Andy (AVD, not AGM) can be consulted.

Also, if all else fails, there is a class of human beings known as "lab bums". These strange creatures can be identified by the fact that they eat, sleep, watch TV and read comic books in the lab, even if they have nothing to do there. Every now and then they may be able to help one with an especially vexing problem, especially if it involves JCL.

Criticisms and Comments

A.M. 101, as it has in the past nine years, runs on your comments. All of those running the course are confirmed card-carrying egotists, and unless you tell them otherwise, they will think the course is running smoothly and will continue their autocratic methods. However, by some strange quirk, they don't mind receiving criticism and will try to change the course to fit any valid criticism.

Comments on assignments and general organization can be given to any grader. Comments on particular lectures should be given to the person who delivered the lecture in question. If your criticism fails to bring desired changes or an explanation of current policy within a reasonable time, feel free to complain to AVD.
In past years some people have occasionally been reluctant to voice constructive criticism concerning the course - this year we would like to see you all take advantage of the opportunity to influence the way things are done.

CONSULTING

Grader consulting hours will be posted at the beginning of each week, and will probably be held at 151 Hope Street, affectionately known as the Software Engineering Complex. Furthermore, the graders have offices there, and can probably be found there when not in class or out drinking. The phone there is x3039.

Also, the Grader's home phones are:

Andy Malis - x4500
Dave Taffs - 272-7094
Brad Silverberg - x4473
Jens Dill - x6212

The above home numbers are for emergencies and nubile young females only.

---

3CJM.
ARRANGEMENTS

TIMES_AND_PLACES

Typically we will have three regular hour-and-twenty-minute classes a week on Monday, Wednesday, and Friday at 3:40. The classes will be held in Barus and Holley 166.

You can be fairly confident that classes will end promptly at 5:00 P.M. If a class begins to run over, simply whisper to the closest 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 get up, YAWN, and leave.

Occasionally there will be optional help sessions dealing with specific topics not covered in detail in class but about which some students feel they need (more) help. At any time three or more students may ask for a help session on any course-related topic. These will most likely be held at night.

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. Most important announcements will be handed out in a blurb, but it's also a good habit to ask a friend what happened in class. Graders are understandably irritated by questions which reflect the fact that the student hasn't read the handouts.

ACCOUNTS

Each of you will be given an account number, good for batch and normal priority interactive use. You are expected to do all of your programming assignments using this number. Please keep in mind that the 360/67 is often saturated and we are on a very limited computing budget - economy and conservation of resources are essential in your computing.
Initially your account will contain $10 - when it begins to run low, see AGM with a reasonable accounting of your spending, and he will put some more in. Remember to conserve runs and debug more than one thing at a time; you are limited to a total of $250 for the semester. If you run over this limit, you will have to hand in your unfinished work with all due consequences - we won't have the money to give you.
The Brown University Computing Laboratory is run as an independent service facility, reporting to the Provost, not to the Division of Applied Mathematics. Its facilities are for the use of all the departments in the university. Students in A.M. 101 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 not ask for nor expect special rights and 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 into 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 just a lousy student and you're doing important research. Yessir.

The second group of people is the LAB STAFF, at least one of whom is an old 101 graduate. Rumor has it that this group is basically friendly to 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. However, you better not bother them too much, or pay very much attention to their predilections, as you yourself may be one some day.

Our people can usually be found in the BUGS room or on a terminal someplace. A good laugh can be had while waiting for output by standing outside the BUGS room and listening to the members of the graphics project talking their strange language. 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 is the best to get us all thrown out of the lab.
OTHER THINGS

COLLABORATION

Due to the nature of your work, 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 and will result in dismissal. We realize the obscurity of the term "undue" - if you are at all unsure of discussing some topic with a fellow student, come discuss it with a grader.

DOCUMENTATION

As in A.N. 51 and 100, we expect detailed structured documentation with each structured program, including a well-written ELD and a detailed run history, with a description of each run made and the cost thereof. In addition, copious comments should be present in the source code itself. Early in the course you will be given a description of the documentation we require. Documentation should be informative and concise, but a little levity or humor is allowed (in the ELD only).

LATENESS

You are expected to turn in assignments on time, and lateness will be heavily penalized. All of the course assignments have been handed out in past years, and we know what are reasonable amounts of time and numbers of runs to spend on each one. Each assignment will have a specific due date, and whatever is completed by then must be turned in. If it is not complete, partial credit will be given, but only if the submission is scruitable and neat, and is "gradeable", i.e., understandable to the graders. Extensions will be given in only "extreme" circumstances, and must be obtained in advance and in writing from AVD or AGN.
MANUAL_FEES

There are three manuals you will need to purchase for the course: IBM PL/I Reference Manual, IBM System/360 Principles of Operation, and IBM Assembler Manual. You may also find it handy to buy an IBM System/360 Reference Data Card (green card). All of the above can be found at the Bookstore. In addition, there will be a $10 fee charged to all students covering the cost of XEROXing several hundred pages of notes, assignments, announcements, etc. This money should be paid to a grader as soon as possible.

SPECIAL_GOODIES

As you get on through the course, you may find yourself becoming especially interested in some aspect of Computer Science. It is highly probable that one of the projects in which AvD and his peons 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).
ORDER OF ASSEMBLER TOPICS

S/360 Fundamentals [pp. 7-15, 24-40 Principles of Operation (POP)]:
- Data types and alignment
- Registers
- PSW
- Instruction formats
- Address Calculation
- Fullword arithmetic
- Halfword arithmetic
- LA
- EEDIT, PACK, UNPACK

Assembler instructions [pp. 19-58 Assembler Manual]:
- Assembler format
- DC, DS, Constant types
- USING, DROP
- START, CSECT, DSECT, ENTRY, EXTRN
- EQU, ORG
- Literals and LTORG
- END
- EJECT, SPACE, TITLE

Branches and condition code [pp. 62-67 POP]:
- BC, BCR
- Extended mnemonics
- BCT, BCTR
- BXH, BXLE
- BAL, BALR

Character string operations [pp. 51-61 POP]:
- SS Instructions
- SI instructions
- IC, STC
- EX, TR, TRT

Subroutines
- Linkage conventions
- Save area chaining
- Register allocation

Decimal and floating point [pp. 35-50 POP]:
- AP, SP, MP, DP, ZAP, CP

-11-
ORDER_OF_LOGICAL_DESIGN_TOPICS

Boolean algebra
Analysis and synthesis of combinational circuits
Karnaugh maps and simplification
Encoders, decoders, and multiplexors
Busssing circuits (data distribution)
Sequential (time dependent) circuits
Logic families and flip flops
Registers, counters, shift registers, and adders
Register transfers
Basic architecture
Data and control flow
Minicomputer architecture
Microprogrammable computer architecture
Physical and logical I/O, channels, and cycle stealing

ORDER_OF_PL/I_TOPICS

Universal language, defaults and hidden facilities,
implicit conversions, machine independence, use of PL/I
(F) compiler options, other versions of PL/I, PL/C, IBM
Optimizing and Checkout Compilers, non-IBM PL/I.

Debugging [pp. 214-237 Ref]:
Structured programming, modularity, symbolic debugging,
error messages, program correctness.

Stream I/O [pp. 99-116 Ref]:
Data, list, edit, formats, files.

Record I/O [pp. 117-138 Ref]:
READ, WRITE, REGIONAL, ENVIRONMENT attribute, INDEXED,
OPEN, CLOSE.

Compiler-time preprocessor [pp. 205-213 Ref]:
Preprocessor variables, expressions, procedures, DO
groups, includes, DEACTIVATE and ACTIVATE, uses,
limitations.
Structures and defined variables [pp. 52-54, 328-331 Ref]:
REFER option, by name, overlay, correspondence, ISUB variables, qualified names.

Runtime environment, data mappings, linkage between PL/I and assembler, organization of the compiler.

Conditions and ON units [pp. 171-177, 308-320 Ref]:
Enabled, disabled, CHECK, SUBSCRIPTRANGE, STRINGRANGE, AREA, user conditions, SIGNAL statement, pseudo-variables.

Conversions [pp. 270-280 Ref]:
Implicit, parameter passing, explicit, precision.

Advanced features [pp. 282-307 Ref]:
VERIFY, TRANSLATE, ALL, ANY, UNSPEC, ADDR, EMPTY, ALLOCATION, etc.
ASSIGMENTS

The following is an overview of the assignments currently planned for 101.

1) Addition of Two Numbers - Add two binary numbers and convert the result to characters (This program will be written in Machine Language, it should take about 12 statements, and you will be allowed 5 days).

2) PL/I Index Function - Implement a simplified version of the PL/I index function (Assembly Language, 2 weeks, 60 statements).

3) Compress - Move data from input area to output areas, packing multiple blanks into a small number of characters (Assembly Language, 3 weeks, 120 statements).

4) Dijkstra's Algorithm - Implement the two pushdown stack version of Dijkstra's Algorithm for parsing arithmetic and logical expressions (PL/I, 1 week, 75 statements).

5) Information Retrieval - Construct a system capable of handling several different types of retrieval requests concerning a small data base. This program will be done by teams of two students, much like Sexatrac (PL/I with Assembly Language, 5 weeks, 600 statements).

In addition, there will be a number of small written and/or programming assignments dealing with the logical design part of the course that have yet to be fully planned, but they will generally overlap with the above assignments. Because of this, it is suggested that you begin work on an assignment immediately after it is given*. More will be said about these assignments later.

*Remember what happened with SORT in AM 100!
MEDICAL DISCLAIMER

CAUTION! This course may become habit forming. Loss of nights' sleep, weekends, and/or vacations may occur. All the graders and lecturers have ulcers anyway, so what the hell...