

MICROCOMPUTER NEWSLETTER

JANUARY 1987

VOLUME 3, NUMBER 3

87 CONTENTS 87

NEWS AND ANNOUNCEMENTS:

Software Development Services for Faculty	19
Zenith Z-181 Laptop	24
IBM PC Users Group	24
Project Assist Workshops	24
Student Programmers Needed	24
Full-Time Programmer Needed	25
Microcomputer Lab Policies	25

REVIEW:

FullPaint	20
-----------	----

SURVEY:

Macintosh Pascal Wars	20
-----------------------	----

BEGINNERS' NOTEBOOK:

When and how do I make backup copies of my work?	22
What files should I have on my disks?	23

SHORT COURSES:

Winter Quarter 1987	26
---------------------	----

NEWS AND ANNOUNCEMENTS

● SOFTWARE DEVELOPMENT SERVICES FOR FACULTY

The Microcomputer Systems Group provides a variety of services for University of Minnesota faculty. Currently, there are quite a number of faculty members in the process of creating instructional software. This instructional software may be written independently or as part of IBM's Project Woksape or Apple's Project MinneMac.

**We have a support center
for faculty members who are
developing instructional software.**

If you are a faculty member who has never developed instructional software before, you will probably find the process both exciting and a bit confusing. The Microcomputer Systems Group can offer you two types of help.

First, you may have questions about how to *design* your software. Perhaps you aren't sure what the characteristics of good instructional software are. You can get help answering these questions through Project Assist. The Project Assist staff offers short courses covering the design of instructional software. They would be happy to help you with the design of your software. You can call them at 626-1090.

Eventually you will reach a point where you know what you want your software to do, and you will feel confident that you have a reasonable design. Once you reach this point, it is time to stop *designing* and start *implementing* your software. This is where our Faculty Support Center enters the picture. While the Project Assist staff's concern is software design, the Faculty Support Center's concern is realizing the design in a usable program. We provide several services to help you get your program written and running. Those services are discussed below.

✓ We have a variety of software tools for both the IBM and Macintosh. These tools simplify and speed the task of writing instructional software. For instance, the MEWS package for the Macintosh can ease the transition from traditional programming to the Mac's event-driven style of programming. Another example is the

The *Microcomputer Newsletter* is published monthly by the Microcomputer Systems Group, a part of the University of Minnesota's Academic Computing Services and Systems department (ACSS).

The *Microcomputer Newsletter* was produced on an Apple Macintosh Plus running MacWrite, MS Word, FullPaint, and PageMaker software; additional pictures by Mac-the-Knife and MacArt Department. Camera-ready newsletter copy was produced on an Apple LaserWriter Plus.

The University of Minnesota adheres to the principle that all persons should have equal opportunity and access to facilities in any phase of University activity without regard to race, religion, color, sex, national origin, handicap, age, or veteran status.

Direct comments, subscriptions, and subscription changes to the editors at the address on the closing page. Subscriptions are free. Permission to copy for noncommercial purposes is granted, provided proper acknowledgment is given; the editors request a copy of the document or record in which our material appears.

REVIEW: FULLPAINT




Reviewing a drawing package often seems more like play than work. Experimenting with FullPaint for the Macintosh definitely seemed like play. FullPaint is similar to MacPaint in that it can do everything that

MacPaint can do. FullPaint differs in that it lets you work on multiple documents simultaneously, has scroll bars, lets you select and copy an entire document, and lets you edit brush shapes. This review will discuss those significant differences.

You can use FullPaint with either a 512K Macintosh or a Macintosh Plus. If your Macintosh has only one single-sided disk drive (400K), you may want to remove some of the fonts, desk accessories, or extra printer drivers from your system disk.

Windows

FullPaint lets you use the entire screen for drawing. This is a great feature if you're working on a big image. To manipulate the drawing window you can use a pull down *Window* menu or keyboard shortcuts. If the tool and pattern palettes obstruct your view (and you are not using the text tool), you can press the <space bar> to hide them or drag them to a different position on the screen. With one click on the FullPaint tool palette, you can hide everything and expand the drawing window to fill the whole screen. You can selectively make the menu bar and palettes reappear.

Not only can you draw over the entire FullPaint screen, but you can also have up to four documents open at once. You can even leave desk accessory windows open while you work. This can be very helpful if you are cutting and pasting between several FullPaint documents or the  Scrapbook.

Scroll Bars

FullPaint has scroll bars along the right side of its window (like MacWrite) and also across the bottom of the window. This is a wonderful feature. Unlike MacPaint, you can move around the FullPaint window without having to select the hand icon from the tools palette. The scroll bars are especially useful when you are in FullPaint's *FatBits* mode. Whereas MacPaint forced you to select the hand icon to move the drawing window, FullPaint lets you select a *FatBits* AutoScroll option. AutoScroll automatically scrolls the window when you reach the edge of the drawing screen. With FullPaint you avoid jumping back and forth from the drawing tools to the hand tool each time you reach the side of the window.

Brush Shapes

MacPaint users know they can edit a pattern. FullPaint gives you more control; it lets you edit brush shapes. You can create a new FullPaint brush shape by holding down the command key while selecting a brush shape from the *Goodies* menu. If you save your changes, you can still revert back to the original.

Printing

FullPaint has more printing options than MacPaint. You can print just a selected area or the entire FullPaint document on either the Imagewriter or the LaserWriter. If you are using a

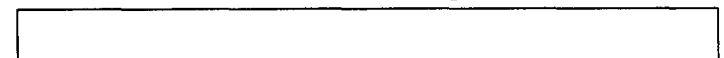
LaserWriter, you can shrink or enlarge the document anywhere from 25-400 percent. With the Imagewriter you can print your document at 50 percent of its original size or expand it 200 or 400 percent.

The Manual

The MacPaint manual, while easy to understand, leaves out a lot of information about using the package, including some of the major short cuts. FullPaint, on the other hand, covers each step and each shortcut. The manual is easy to follow and is loaded with excellent information, including plenty of examples to give you a visual understanding of how to use FullPaint.

Conclusion

We found FullPaint a definite improvement over MacPaint. With its additional features, FullPaint is actually easier to use than MacPaint. You can purchase FullPaint at the Book Center in Williamson Hall for \$63. If you would like to take a closer look at FullPaint, we have it at the Microcomputer HelpLine.



SURVEY: MACINTOSH PASCAL WARS



Until recently, if you wanted to use the Pascal programming language on the Macintosh you did not have many choices. In the last few weeks this situation has changed dramatically. Both Turbo Pascal and Lightspeed Pascal are now available, and at least one more Pascal implementation that we have not yet seen is on the way: Apple's Macintosh Programmer's Workshop (MPW). In this survey, we will look at how Turbo Pascal and Lightspeed Pascal compare with Mac Pascal and TML Pascal, two products that have been on the market for many months. In addition, we will offer some thoughts about which Pascal you might want to buy if you have a Macintosh. As you read this survey, bear in mind that, as of this writing, we have only used Lightspeed Pascal for three weeks and Turbo Pascal for two weeks. After we have more experience with these products we plan to run an in-depth review.

Macintosh Pascal

In the beginning, there was Macintosh Pascal. One of Mac Pascal's virtues continues to be its price; the Book Center sells a student version for \$33. This student version does not include the Mac Pascal Technical Reference manual, but otherwise it is the same as the regular version which sells for \$59 at the Book Center.

Macintosh Pascal has another virtue: Mac Pascal is an *interpreted* version of the Pascal language. Interpreted languages translate program statements into executable code while the program is running. Because Mac Pascal is an interpreted language, you can single-step through your program and watch how the values of your program's variables change after each line in the program is executed. Interpreted languages have a vice: their programs run slowly. A significant amount of overhead is involved in translating each instruction of a program each time the

instruction is executed. An interpreted program's overhead typically means that the program will run 5-10 times slower than one generated by a *compiler*. A *compiler* translates the program statements into executable code before the program runs. Compiled programs have less overhead and so run quite quickly.

TML Pascal

TML Pascal was released about a year ago. This event was cause for celebration in the Macintosh community because TML is a Pascal compiler. Even with the first version of TML, you could write commercial-quality programs that ran very fast. There have been several new versions of TML Pascal since the version 1.0 release. Since January 1986 we have seen versions 1.1, 1.2, and 2.0, and there are rumors that version 2.1 is coming soon. Because TML was the only Pascal compiler available for nearly a year, there are a variety of tools that are available only to TML programmers. In fact, the Microcomputer Systems Group wrote a set of tools for TML Pascal known as MEWS. (MEWS is available at the Microcomputer HelpLine; bring two blank initialized disks and you can copy MEWS.)

TML Pascal has some drawbacks. The editor that you use to write your programs is separate from the compiler. This means that unless you write perfect programs the first time, you will spend a considerable amount of time jumping between the compiler and editor as you modify your program. Before you can use TML for the first time, you must tell it where its libraries are located; so you must specify the library's complete path name (the disk name and folder names). Path names can become quite long if you have a hard disk and store TML in a folder. If you write textbook Pascal programs (programs that do not use the Macintosh user interface), you will find that you cannot backspace to correct errors when your program reads standard input using `readln` or `read`. Most Pascals allow backspacing to correct typing errors for textbook Pascal programs.

TML Pascal has redeeming features. Apple's *Inside Macintosh* documentation covers the Macintosh ROM and system routines in great detail. Apple documented the Mac's ROM routines in Pascal, so it is nice if your Pascal compiler adheres to Apple's documentation. Every routine documented in *Inside Macintosh* that we have tried works (as documented) in TML Pascal. More importantly, TML seems to be committed to keeping TML Pascal compatible with Apple's definition of Pascal. Apple's forthcoming MPW Pascal will include extensions for object-oriented programming. Version 2.0 of TML Pascal already includes these object-oriented extensions.

Until recently, we usually recommended TML Pascal to serious Macintosh programmers. TML (\$59 at the Book Center) is still a reasonable choice, especially if you are interested in object-oriented programming (neither Turbo Pascal nor Lightspeed Pascal have object-oriented Pascal extensions), and you do not plan to write textbook Pascal programs.

Lightspeed Pascal

Lightspeed Pascal is a new product from the same people

(THINK Technologies) who wrote the interpreted Macintosh Pascal. Lightspeed Pascal is an attempt to keep the virtues of an interpreter (the ability to single step through a program and view program variables while running the program) and to eliminate the vice associated with interpreters (slow program execution). Lightspeed is moderately successful at achieving these goals. Lightspeed Pascal produces compiled programs, so the programs are considerably faster than Mac Pascal programs. Writing a program with Lightspeed is fairly fast because the Lightspeed editor and compiler are one unit. Since the editor and compiler are integrated, there is little delay when jumping from editor to compiler and back. When the Lightspeed compiler finds a syntax error in your program, it automatically jumps to the editor at the point where the error occurred. This is an improvement over TML Pascal where the editor and compiler are completely separate.

Yet, we are unhappy with several features of Lightspeed. The people who developed Lightspeed went to great lengths to provide debugging information; it has a debugger package named Lightsbug. Although you can use Lightsbug to view the variables in your program, the values of the variables are displayed in hexadecimal. It seems to us that if someone goes to the trouble of writing a debugger specifically for a compiler, we should have the option of viewing the values of variables in a form other than hexadecimal. (For example, if the variable is a real number, we would like to display the real number rather than hexadecimal digits.)

The editor portion of Lightspeed is very similar to the editor portion of Mac Pascal. That is, the editor formats your program statements automatically while you edit. This feature can be irritating because in the course of writing a program you frequently have incomplete program statements. Lightspeed's editor flags incomplete statements by placing them in an outlined font. The effect of an editor that continually flags incomplete statements is similar to that of a nagging younger brother: while he may be right, you don't want to hear about it all the time. The overhead involved in automatically formatting program statements also causes Lightspeed's editor to be a bit sluggish.

This brings us to the root of our complaints about Lightspeed. While the people who wrote Lightspeed did some amazing work, their work may have been misdirected. Since you spend most of your time working in the editor when you write programs, the editor should be as fast and responsive as possible. Also, whenever you aren't editing, you're waiting for the compiler, so the compiler should be as fast as possible. The greatest productivity gains can be realized by speeding up the editor and compiler so that the edit/compile cycle is fast. Rather than optimizing the speed of the editor and compiler, Lightspeed attempts to make their compiler act like an interpreter. Not surprisingly, Lightspeed was only actively used for one week by programmers in the Microcomputer Systems Group. The next week Turbo Pascal arrived.

Turbo Pascal

Turbo Pascal for the Macintosh was written by Borland

International, the same people who wrote Turbo Pascal for the IBM. The philosophy behind both Turbo Pascals is the same: make editing and compiling programs *fast*. Borland succeeded: Turbo Pascal for the Macintosh is amazingly fast. *It takes 3.5 seconds to compile a 700 line program to memory with Turbo Pascal on a Macintosh Plus.* We ran a series of tests using a Mac Plus with an Apple SCSI hard disk. The results are summarized below:

	compile time (seconds)		
	to Memory	to Disk	and Launch
Turbo	3.5	5.0	4.0
Lightspeed	8.0	15.0	13.0
TML	-	16.0	48.0

Turbo Pascal's fast compilation time helps speed the edit/compile cycle. The compiler is also integrated with the editor, so when the compiler finds a syntax error in your program it returns you to the editor at the point where the error occurred (as is the case with Lightspeed Pascal). The editor is also very fast and responsive. You can quickly scroll from one part of your document to another, and you experience none of the annoying delays that plague Lightspeed's editor.

With Turbo Pascal you can compile textbook Pascal programs or programs that use the full Macintosh interface. When compiling textbook Pascal programs, the Macintosh version of Turbo is source-code compatible with the IBM version of Turbo. This means that you can compile and run programs written for the IBM (in most cases) without modification on the Mac. The only potential problem is if your program directly addresses the IBM hardware; in this case you will need to rewrite your program because Macintosh hardware is different from IBM hardware.

Turbo Pascal for the Macintosh provides full access to the Apple ROM and system routines. In most cases we moved programs that were originally written with TML Pascal into Turbo Pascal in less than 15 minutes. A question that comes up when one compiler is much faster than the other is: How good is the code produced by the compiler? The jury is still out on whether complex programs compiled with Turbo run as fast as programs compiled with TML or Lightspeed. We timed a simple program that initializes and inverts a rectangle on the screen 5000 times. The results are summarized in the table below:

	run time (seconds)	code size (K)
Turbo	46	14.5
Lightspeed	47	9.0
TML	46	8.5

For this simple test, all the programs ran at about the same speed; but, as you can see, there were significant differences in the size of the program file.

We plan to use Turbo for most of our program development efforts. We believe the time saved by Turbo's fast edit/compile

cycles makes Turbo Pascal a reasonable choice even if you plan to compile the final version of your program with another compiler. Moreover, the Turbo and TML compilers are similar enough that moving a program from one to the other is not a difficult task.

The main criticism we have concerning Turbo Pascal is that the Macintosh's version of the manual is like the IBM manual. The information in the manual could be better organized. Consequently, it can be difficult to find what you are looking for in the manual. On the other hand, this hasn't kept Turbo from being extremely popular in the IBM world. We expect the same popularity for the Macintosh version. As you may have guessed, we have moved our programming tools for TML Pascal (MEWS) over to Turbo. For the foreseeable future we plan to support both TML Pascal and Turbo Pascal. (The Turbo version of MEWS is available at the Microcomputer HelpLine. Bring two blank, initialized disks and you can copy it.) Turbo Pascal for the Macintosh is \$60 at the Book Center in Williamson Hall.

Summary

Selecting a Pascal compiler depends on what kinds of programs you want to write and how you write (and debug) your programs. We feel that Turbo Pascal's fast edit/compile cycles make it a good choice for most people. TML Pascal has the virtue of having a variety of support libraries as well as object-oriented programming extensions. Macintosh Pascal's main virtue is the student version's low price. Lightspeed Pascal attempts to incorporate the advantages of an interpreter in a compiled language, but the overhead involved makes Lightspeed relatively slow for developing programs.

BEGINNERS' NOTEBOOK



This month we are starting a new series of articles for the novice microcomputer user: the *Beginners' Notebook*. This month we will cover two topics that every beginner needs to know about:

1. When and how do I make backup copies of my work?

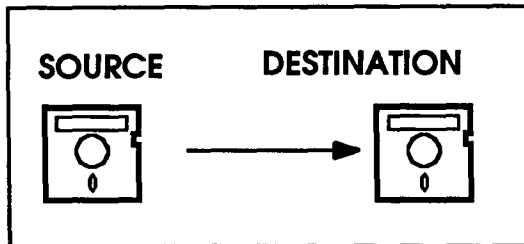
Everyone tells new computer owners that they should make plenty of backup copies of their work. But *why* should you make backups and exactly *how* do you do this? During the excitement of the first few months with a new microcomputer, it is easy to forget the ugly reality of computing: *eventually, whatever can go wrong will go wrong*. While there are many different things that *could* go wrong, the end result is usually that you cannot read or retrieve your work from a disk. Think about this for a minute: if you can't read your work from the disk on which it is stored, your work is *lost forever*. This means you will have to recreate the work you have done.

Because you cannot predict when a problem will occur, the only way you can avoid losing work is to make duplicate copies of *all* your work *all* the time. Then, when a problem does occur, you

can resurrect your work by reading from one of the duplicates. Obviously, duplicate (or backup) copies of your work are essential if you are working against a deadline. Having only one copy of a report that took you days to create (and is due in four hours) is an invitation to disaster.

How do you copy your work? The details depend on which microcomputer you use, but no matter what microcomputer you are using, you should keep the copies of your work on one (or more) *separate disks*. It is prudent to keep the backup copies of your work on a different disk from the original (the source) so that you can use the backup copy even if the entire disk that holds the original is destroyed.

So, to backup your work, you will want to copy the file (or document) that contains your work from one disk to another.



✓ Macintosh

If you are using a Macintosh, you can copy files from the desktop. To copy from the desktop, you should put the disk onto which you will be copying (the destination disk) into the machine. If you have two disk drives, insert the *destination disk* in one drive and the *original disk* in the other drive. If you have a single drive Mac, eject the original disk so you can insert the destination disk in the internal drive. Next, tell the Mac to copy the file. Do this by using the mouse to position the pointer (the arrow) on top of the icon for your document. Hold down the button on the mouse and drag that document's icon to the icon for the disk onto which you want to copy the document. For further details see your manual.

✓ IBM-PC or compatible

If you have an IBM-PC (or compatible), you can copy the file containing your work to another disk by using the DOS COPY command. Suppose that the file that we want to copy is named FRED.WRK and this file is on the floppy disk in drive A:. To copy this file to the floppy disk in drive B:, you would type the following:

```
COPY A:FRED.WRK B:
```

For further details see the section covering the COPY command in your DOS manual.

2. What files should I have on my disks?

Shortly after you discover that you can copy files from one disk to another, the question of which files need to be on a disk comes up. There aren't any rules about which files must reside on a disk, so you have quite a bit of freedom. The files you will want to put on your disk depend on what you plan to do with the disk. If you want to be able to use the disk to start your machine (or *boot* the machine, if you like jargon), you need to have the operating system on the disk.

✓ Macintosh

If you want to start up your Macintosh from a disk, you must at least have the FINDER and SYSTEM files on that disk. You usually find these files inside the SYSTEM FOLDER. If you want to print on the Imagewriter printer, you also need to have the IMAGEWRITER file on your disk; this file holds the instructions that the Mac uses when printing on the Imagewriter. If you want to print with the LaserWriter, you need two files on your disk: LASERWRITER and LASERWRITER PREP. Most people will also want the SCRAPBOOK file on their disk (this file is where the Mac stores the contents of the scrapbook).

✓ IBM-PC or compatible

On the IBM-PC (and compatibles) the DOS operating system can be copied onto a disk when you format the disk (see the FORMAT command in your DOS manual for details). Since many of the DOS commands are *not* built into the operating system, you may want to copy the files that contain specific commands onto your disk. For example, the DISKCOPY command is not built into the operating system; it is stored in a file named DISKCOPY.COM. If you want to execute the DISKCOPY command, you must have this file on your disk. It is easy to find files that contain commands because there is a convention for naming command files: the file name must end in either COM or EXE. Here are two examples

```
TREE.COM  
WP.EXE
```

If you look at the DOS disks that came with your machine, you can see that they have many command files. Since these disks contain *everything*, surely there are some command files that contain commands you don't use or rarely use. You can save space on your working disks if you don't copy these files.

What will happen if you need one of these commands? If you type a command and the file that contains the command is not on your disk, the IBM-PC will respond with the message: BAD COMMAND OR FILE NAME. This message implies that you either mis-typed the command, or the file that contains the command is not on your disk. If this happens, you can copy the command file from the DOS disks that came with your machine onto your working disk.

✓ Storage Space

Whether you have a Macintosh or an IBM, there is no reason to put the operating system software onto a disk if you don't want to use that disk to start your machine. Since the operating system requires a considerable amount of disk space, you may want some disks without an operating system so that you have more room for storing documents. A common strategy for organizing disks is to put the operating system and the application you want to use, for example, your word processing program, on the disk that you use to start your machine. In the other disk drive you have a disk without the operating system or word processing program. You use this other disk to hold the documents that you create. Since this disk contains *only* documents, it has plenty of room for saving your documents.

Announcements continued from page 19 — — —

CATS package for IBM personal computers. CATS is a simple, easy to use software authoring system.

✓ We are running a series of Apple and IBM Developers' Roundtables. These Developers' Roundtables will keep you informed of known bugs in programming languages as well as new software tools and programming techniques. At these roundtables we also seek feedback from you about the kinds of programming tools you want. Project MinneMac and Woksape participants receive notification of the Developers' Roundtables in the mail. If you are working on an independent project and are interested in attending these roundtables, send us your mailing address and tell us which roundtable interests you. (Our address is listed below.)

✓ If you are a participant in Project MinneMac or Project Woksape and need assistance writing your program, we have some limited programming support available. Contact the technical coordinator for your project to explore the possibilities for programming support.

The phone number for Earl Schleske, technical coordinator for *Project Woksape*, is 625-5501.

The phone number for Mark McCahill, technical coordinator for *Project MinneMac*, is 625-5501.

✓ Whether you are writing your program yourself, having a student write it, or have hired a consultant, we will be happy to look at the program and suggest coding and debugging strategies.

We hope that you will find our Faculty Support Center a useful resource. Our campus mailing address is: Microcomputer Systems Group, Faculty Support Center, 125 Shepherd Labs.

● ZENITH Z-181 LAPTOP

We gave you our first impressions of Zenith's lightweight (less than 12 pounds) laptop with the amazing display in our *August 1986* newsletter. At that time we could not tell you the University price nor when we would have a machine for you to try. Finally, we have a price, and you can see the Z-181 in our HelpLine and form your own impressions.

For those who missed our *August 1986* newsletter, here's a description of the Z-181. The Z-181 is considerably different from Zenith's other portable, the Z-171. The most significant differences are the 3.5" disk format and the liquid crystal display's (LCD) supertwist technology. The supertwist LCD has a much higher contrast ratio than a conventional LCD. In addition, the display is backlit, so it does not depend on ambient room light; and its aspect ratio makes a circle look like a circle instead of a squashed ellipse. The combination of backlighting and the super twist LCD gives the Z-181 display a contrast ratio of about 12:1. Conventional LCDs have a contrast ratio of about 4:1. The Z-181 has a CMOS 8088 Central Processing Unit that runs at 4.77 MHz. The battery works up to five hours before it needs

recharging, and an A/C power adapter is standard (so you can use the Z-181 on regular household electricity.) Although the Z-181 has a socket for an 8087 numeric coprocessor, we don't have an 8087 in the machine we have in the HelpLine. (Battery life is very short when the 8087 option is installed.)

You can place an order for the Z-181 at the Book Center in Williamson Hall. The University's configuration includes 640K RAM, one serial (DB25) and one parallel port, two shock-mounted 3.5" 720K floppy drives, a 9-pin RGB connector (so you can connect an external color monitor), and built-in 25-line by 80-column display and keyboard. The Z-181's price is \$1460 plus shipping. The Book Center does not have prices yet for any of the Z-181 accessories, such as the Hayes-compatible modem, external 5.25" floppy drive, or external battery pack recharger.

● IBM PC USERS GROUP

The next IBM PC Users Group meeting will be held on Thursday, January 29th, on the East Bank Campus in the Architecture Building, Room 25. The meeting will begin at 3:30 p.m.

● PROJECT ASSIST WORKSHOPS

Project Assist is a campus-based group that helps University of Minnesota faculty who want to use computers for instruction. Project Assist is offering the following workshops during Winter Quarter.

- *Introduction to PLATO*, January 15, February 6, and March 3 in the Appleby Computer Lab

The following workshops are held in 335 Peik Hall.

- *Evaluating Authoring Systems*, January 14
- *Computer-Based Instruction in Higher Education: A Guided Tour of Possibilities*, January 28
- *Evaluating Instructional Software*, February 3
- *A Programmer's Guide to Instructional Software*, February 19

For more information about the workshops call Paul Wieser, Project Assist, at 626-1090.

● STUDENT PROGRAMMERS NEEDED

Occasionally our HelpLine receives requests for referrals of student programmers. Typically, these requests come from University faculty and departments that have software projects they need some programmers to complete. The Microcomputer Systems Group has established a database of University of Minnesota students who are interested in this type of work.

If you want to be included in this programmer pool, visit the Microcomputer HelpLine and fill out our questionnaire. Note: we do not guarantee work. We simply add your name to the pool of student programmers who are interested in work.

● FULL-TIME PROGRAMMER NEEDED

The Microcomputer Systems Group has an opening for a full-time programmer. The person in this position will specialize primarily in Macintosh programming. This is a civil service position; the job title is Analyst/Programmer. A BS in Computer Science (or equivalent) plus two or more years of programming experience with Pascal and Motorola 68000 Assembly Language and knowledge of TML Pascal for the Apple Macintosh is required; knowledge of the Unix operating system is desirable.

Apply for this position through the University's personnel department. When you apply for this position you will need to know the Job Requisition Number; the number is 053469. The monthly salary range is \$2031-3236, and the starting date is ASAP.

● MICROCOMPUTER LAB POLICIES

Since some of our readers may be unaware of the Microcomputer Labs referred to on the right, we are providing a little background information.

The Microcomputer Access Card costs \$20/quarter. The owners of this card can use the equipment in the eight microcomputer labs overseen by Information Systems. The individual labs are managed by different departments. The labs have equipment such as IBM and Zenith personal computers, Apple Macintoshes, Apple IIs, and printers. For information about the specific hardware and software any one labs has and for hours of operation, you must call the individual labs. The labs' locations and phone numbers are printed on the back of the Access Card.

ACSS is one of the departments that manage some of these public (public to the University of Minnesota community) microcomputer labs. The ACSS labs' locations and phone numbers are:

14 Folwell Hall, 625-4896
306B Lind Hall, 625-9032 and
9 Walter Library, 626-1899.

The other five labs are managed by several different departments. The locations and phone numbers for those labs are:

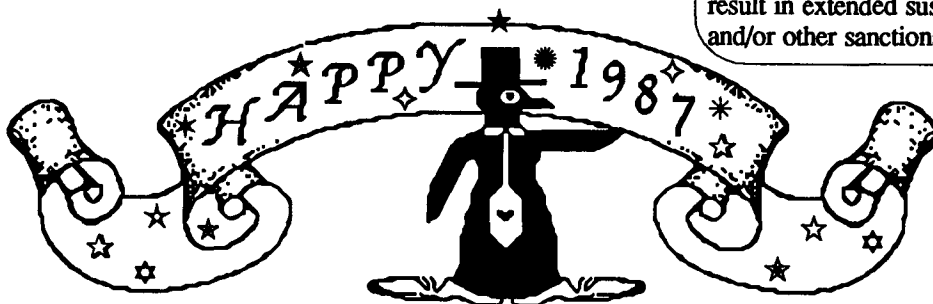
8 Moos Tower, 625-1477
54 Eddy Hall Annex, 625-0314
204 Appleby Hall, 625-5080
170 Anderson Hall, 624-6526 and
B50 Central Library, St. Paul Campus, 624-3269.

MICROCOMPUTER LAB ACCESS AND USE POLICIES

The Microcomputer Access Card affords access to ACSS's and other microcomputing labs for University students, faculty, and staff. The card itself is clearly marked *non-transferrable* which means that it is to be used only by the original purchaser and that person's signature and ID stamp should appear on the card (it will be stamped the first time it is presented in a lab). Access to lab facilities by individual Microcomputer Access Card holders may be restricted or denied for the following reasons.

1. Unauthorized use of a Microcomputer Access Card, including but not limited to:
 - ineligibility (lab users must be University of Minnesota students, faculty, or staff);
 - using a stolen, "found," or loaned card;
 - loaning your own card to someone else to use.
2. Unauthorized changes to lab hardware or software, including but not limited to:
 - disconnecting and reconnecting, or configuring hardware;
 - removing, changing, or reconfiguring files on lab disks;
 - damaging lab hardware or software or removing any lab property from the lab.
3. Failure to observe lab policies, procedures, and protocol, including but not limited to:
 - refusing to sign in and out of the lab properly;
 - moving from the assigned machine onto another without the explicit permission of the lab attendant;
 - refusing to leave the lab promptly at closing time;
 - refusing to respond to, or responding inappropriately to, requests made by the lab attendant (e.g., a request to move to another machine or to delay printing due to a class being held in the lab) in the normal course of carrying out their job responsibilities;
 - using threatening or abusive language or behavior directed at anyone in the lab.

Any violations under Section 1 (above) will result in confiscation of the Microcomputer Access Card(s) involved and suspension of lab use privileges for at least one quarter. A first offense in either of the other two categories (above) may result in confiscation (at the discretion of the lab attendant) of the user's Access Card, pending an administrative review of the matter. Repeated violation may result in extended suspension or denial of lab use privileges and/or other sanctions.



MICROCOMPUTER SHORT COURSE DESCRIPTIONS

IBM

IBM and Compatibles Hands-On: Enrollment Limited to 10 Per Class

Introduction to Microcomputers - DOS. Fee Level B. This 5-hour course for new users is offered over two days; it includes background information on hardware and software for microcomputers as well as a practical (hands-on) introduction to the MS-DOS and PC-DOS operating systems.

Introduction to Microcomputers or equivalent knowledge is required for the IBM courses listed below.

Changed - Managing Your Hard Disk. Fee Level C. In this 4-hour course we demonstrate the basic steps involved in organizing a hard disk. Some of the concepts covered are creating directories and batch files, backing up a hard disk, and transferring data between subdirectories.

Introduction to dBASE III. Fee Level D. This 6-hour course will cover basic concepts of database management. The course is taught over three days and covers only interactive commands. You will create several databases and learn how to enter data, modify it, and retrieve it.

Programming with dBASE III. Fee Level C. In this 5-hour course you will write and run several simple dBASE III programs while learning some basic programming concepts. The course is taught over two days. *Beware.* Working knowledge of dBASE III is required!

Beginning Lotus 1-2-3. Fee Level B. This 2.5-hour course will familiarize users with basic Lotus 1-2-3 spreadsheet concepts. You will be introduced to beginning-level commands by entering a sample spreadsheet. You will set-up your own spreadsheet, enter data, formulas, use spreadsheet commands, functions, formats, and create several charts.

Intermediate Lotus 1-2-3. Fee Level B. This 2.5-hour course covers three of Lotus 1-2-3's more advanced features: database queries, look-up tables, and macros. *Beware.* Working knowledge of Lotus 1-2-3 is assumed. *The Beginning Lotus 1-2-3 class alone is insufficient!*

Introduction to MS (Microsoft) Word. Fee Level C. This 5-hour course for new MS Word users is taught over two days. In it you will learn to use keyboard commands to format characters and paragraphs, move text, use search and replace, multiple windows, a glossary, and create headers and footers. You will install a printer and change printer fonts. We will give a short demonstration on how to use a mouse with MS Word.

Introduction to WordPerfect. Fee Level B. This 2.5-hour course is for new WordPerfect users. You will learn to use simple line and page formatting commands (such as set margins and headers), how to manipulate blocks of text, use search and replace, change initial (default) values, and how to install a printer.

MAC

Prior knowledge and familiarity with the Macintosh is assumed for all these classes.

🍏 **No hands-on: Enrollment Limited to 12 Per Class**

Overview: (Microsoft) Excel. Fee Level A. This 2.5-hour overview consists of discussion and demonstration of Excel. We will demonstrate designing and manipulating simple spreadsheets and databases, graphing data from spreadsheets and databases, and designing and using function and command macros.

Overview: MS (Microsoft) Word. Fee Level A. This 2.5-hour overview will include a discussion and demonstration of the basic features of Word Version 1.05 and how these features work together.

Overview: Preparing a Dissertation on the Macintosh. Fee Level A. In this 2.5-hour overview we will demonstrate and discuss the advantages and disadvantages of several word-processing and graphics programs currently available for the Macintosh. These programs include: MacWrite, MS Word, MacDraw, MacDraft, Chart, and others. We will discuss organizing bibliographies, graphing and organizing data, placing graphs and illustrations into your document, and using the Apple LaserWriter printer.

Overview: Omnis 3. Fee Level A. This 2.5-hour overview is an introduction to Omnis 3, a relational and hierarchical database management program designed for very large databases which require custom-designed data entry forms. In this class we will design and implement a database application. *Beware.* Prior familiarity with a programming language, including loops and IF statements, is assumed!

New - **Overview: Cricket Graph. Fee Level A.** This 2.5-hour overview demonstrates some of Cricket Graph's features and compares them to Microsoft's Chart.

🍏 **Hands-On: Enrollment Limited to 10 Per Class**

Intermediate Excel. Fee Level C. This 5-hour workshop is taught over two days. In it you will use Excel's graphics, macros, spreadsheet, and database features. *Beware.* Prior experience with Excel or attendance in the introductory Excel course is assumed!

Registration: Classes are filled in the order registration is received. Register in person at the ACSS (Academic Computer Services and Systems) Reference Room in 128A Lind Hall, Monday-Friday, 8 a.m. to 4:30 p.m. or by mail.

The deadline for registration is 4:15 p.m. on the last working day *before* the class begins.

Fees: Fees must accompany your registration. You can pay fees by cash, check, or a signed University Journal Voucher. *No refunds will be made after the class has begun.* Fees are printed in order for the following groups:

- ① University students,
- ② University faculty and staff, and
- ③ non-University persons.

Fee Levels:

A = \$15, 25, 40	B = \$25, 35, 60
C = \$40, 50, 80	D = \$45, 60, 90

More Information: If you need more information on these short courses, call Jerry Stearns at 625-1543. For registration information, call 625-7397.

WINTER QUARTER 1987: MICROCOMPUTER SHORT COURSES

KEY: Macintosh classes are marked with 🍏.

IBM classes are unmarked.

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
JANUARY					
	12 1:30-4:00, 1 of 2 Intro • DOS	13	14 1:30-4:00, 2 of 2 Intro • DOS	15	16
	19 University Holiday	20	21	22 9:30-noon Intro • WordPerfect	23
		1:30-3:30, 1 of 3 Intro • dBASE III	1:30-3:30, 2 of 3 Intro • dBASE III	1:30-3:30, 3 of 3 Intro • dBASE III	
	26	27 1:30-4:00, 1 of 2 Intro • DOS	28 1:30-4:00 Begin • Lotus 1-2-3	29 1:30-4:00, 2 of 2 Intro • DOS	30 1:30-4:00 🍏 O'view • Excel 🍏
FEBRUARY					
	2	3 1:30-3:30, 1 of 2 Org • Hard Disk	4 1:30-4:00 Intro • WordPerfect	5 1:30-3:30, 2 of 2 Org • Hard Disk	6
			1:30-4:00 🍏 O'view • MS Word 🍏		
	9	10	11 9:30-noon, 1 of 2 Intro • DOS	12	13 9:30-noon, 2 of 2 Intro • DOS
	16	17	18 9:30-noon, 1 of 2 🍏 Intermed • Excel 🍏	19	20 9:30-noon, 2 of 2 🍏 Intermed • Excel 🍏
	1:30-3:30, 1 of 3 Intro • dBASE III	1:30-4:00 🍏 O'view • Omnis 3 🍏	1:30-3:30, 2 of 3 Intro • dBASE III	1:30-4:00 Intro • WordPerfect	1:30-3:30, 3 of 3 Intro • dBASE III
	23 1:30-4:00, 1 of 2 Intro • DOS	24	25 1:30-4:00, 2 of 2 Intro • DOS	26 1:30-4:00 Begin • Lotus 1-2-3	27
MARCH					
	2	3 9:30-noon, 1 of 2 Intro • MS Word	4	5 9:30-noon, 2 of 2 Intro • MS Word	6
	1:30-4:00 Intermed • Lotus 1-2-3	1:30-4:00, 1 of 2 Prog • dBASE III	1:30-4:00 Intro • WordPerfect	1:30-4:00, 2 of 2 Prog • dBASE III	1:30-4:00 🍏 Cricket Graph 🍏
	9	10 1:30-4:00, 1 of 2 Intro • DOS	11 9:30-noon 🍏 Dissertation 🍏	12 1:30-4:00, 2 of 2 Intro • DOS	13

Return this coupon (or a photocopy) to the address listed below if you want to be added to our mailing list. Please be legible.

MICROCOMPUTER NEWSLETTER MAILING LIST

Name _____

If University of Minnesota Twin Cities Campus Mail:

Do not use a personal address; Campus Mail only delivers bulk-mailed items with an official departmental address.

Department _____

Building _____ Room _____

If U. S. Postal Service:

Address _____

City _____ State _____ Zip _____

(1/87)

University of Minnesota Affiliation. Please. Check only one box.

Department Faculty Staff Student Alumni Other

MICROCOMPUTER SYSTEMS GROUP

phone-in First Aid for microcomputer users since 1980

➔ **Call our HelpLine**
626-4276
(dial MA MICRO)

➔ **Visit our HelpLine**
Room 125, Shepherd Labs
East Bank Campus

➔ **We're Available**
9 a.m. - 12 and 1:30 - 4 p.m.
Monday
Tuesday
Friday

9 a.m. through 4 p.m.
Wednesday
Thursday

➔ **Newsletter Editorial Staff**

Joanne Bergman, Dave Burris, Jung-Juin Chen, Steve Collins, Mary Dickel, Simin Hickman, Dave Larsen, Mark McCahill, Earl Schleske, Polina Shur, Ellen Thayer, Julie Thorndycraft, Shih-Pau Yen

MICROCOMPUTER NEWSLETTER

c/o Microcomputer Systems Group
University of Minnesota
Room 125, Shepherd Labs
100 Union Street SE
Minneapolis, MN 55455

Nonprofit Organization
U.S. Postage
PAID
Minneapolis, Mn 55414
Permit No. 155

UNIVERSITY ARCHIVES

10
WALIB

55455