A course of formal instruction in on-line computing at Carnegie-Mellon University*

LEE W. GREGG

Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

The name of the course is called "Computer Methods in Psychology." That's what the name is called. The name is really "Computers with and without Thought." The course is called "Obscene Things," at least by some students, but that's only what it is called. The course is "A-sitting at the Console," and it is designed to clear up confusion when it is not clear what a thing is called.

Indirect addressing is just one area of confusion, so that our students, I am sure, are often as bewildered as Alice was in her conversation with the knight. Since 1968, our department has offered a course to help students distinguish between "Datasets 1," the files created under the IBM TSS 8.1 operating system, and "Datasets 2," the Bell Telephone devices that we use to dial up our high-speed line connecting the main machine with our laboratory computer.

The course also represents our attempt to deal with computers and programming in psychologically relevant ways and with respectable intellectual content. It is offered to first-year graduate students and advanced undergraduates. Class size is typically small, ranging from 10 to 15 students during the past 3 years. There are two main purposes of the course: (1) to convey basic concepts of programming, and (2) to develop competence in using the computer facilities that are available at our institution.

Contemporary information-processing approaches to psychology are based in part on computer concepts and have adopted some of the computer's terminology. It is necessary to understand these concepts to comprehend the approach. Examples of the more basic concepts are the idea of specifying activity by a program or algorithm, the idea of a conditionally specified action, of a subroutine, of recursion, of data structure, and the distinction between program and data.

These concepts are developed within three distinct contexts during the one-semester course.

(1) *Psychological data analysis* and access to the computation center's IBM 360/67 TSS system. This portion of the course is brief, about 10% of the total time. It relies on user manuals prepared by the CMU computer center staff. The students are expected to learn how to use the TSS editor and command language

to write and execute simple programs in a numeric language. This background is necessary for doing problems assigned in a statistical methods course usually taken concurrently.

(2) *Computer simulation* and an introduction to a simulation language. This section is presented during the last 40%-50% of the semester. Lectures and demonstrations prepare the students to create a simulation model of behavior. This year, the simulation language we are using is SNOBOL4.

(3) *On-line computing* and access to the Computer-Controlled Psychology Laboratory (CCPL). This part of the course is of principal relevance to the present discussion. Let me say a few words about the hardware and software of the laboratory, and then describe briefly the content and methods used.

THE CARNEGIE-MELLON UNIVERSITY CCPL DESIGN DECISIONS

The laboratory is described in greater detail in another paper presented at this conference. The hardware configuration and software organization emerged from three considerations that were resolved during the evolution of the CCPL. First, we wished to create as much flexibility as possible in the ways the on-line computer could be used. We decided that each laboratory room should have a general-purpose interface to the computer rather than allocate special functions or specific devices, e.g., slide projector in one room, Teletype terminals in another, etc. Hence, each room interface is an extension of the I/O bus of the Honeywell DDP-116 computer with 16 bits in and 16 bits out. The implication of this decision for instruction is that a substantial problem of mapping specific devices to computer words must be included as a topic to be discussed and explained. Second, for the purposes that the laboratory was originally intended, we wished to provide a time-sharing environment so that multiple on-line users could conduct experiments simultaneously. The command language for the operating system FIASCO must therefore be taught before the student can gain access to the CCPL computer.

The third basic consideration we faced concerns the extent to which we should provide user consultant support for the CCPL. Our decision was that we did not want to run a computation center. Each user must be responsible for writing and debugging his own program. We do provide technical assistance at the hardware level—primarily to insure that users do not attach inappropriate devices to the room interfaces. And, of course, system maintenance and software support are

*The Computer-Controlled Psychology Laboratory at Carnegie-Mellon University was developed under a research grant from the Public Health Service, National Institute of Mental Health, Grant MH07722.

Preliminary Problem Set:

1. Display your name on the CCI screen.
2. Read into the computer as many characters as necessary for your name and then display your name at a different location on the CCI screen.
3. Blank the CCI screen.
4. Display your name on the CCI screen for a period of time. Then blank the screen for a brief interval.
5. Make your name flash on and off.
6. Cause your name to be recorded on the ASR35 Teletype.
7. Read a number into list NUM1 and another number into list NUM2. Record the sum of these numbers on the Teletype.
8. Measure the time between pushes of any buttons on the I/O panel and record the time on the teletype.
9. Read 5 characters into list STIM. Select each character in turn and display them on the CCI at the center of the CCI screen.
10. Choose one of two alternate lists randomly. The first list should read "WIN" and the second should read "LOSE." Display the list at position 10 15. Wait for a button to be pressed on the I/O panel before repeating.

Fig. 1. The preliminary problem set for introducing students to on-line programming via the APCOL compiler.

centralized. The decision requires that users become proficient in programming and have some knowledge of the object code and machine-level functions of the computer.

The CCPL, therefore, provides an operating system environment with a fairly sophisticated time-sharing monitor with on-line debugging capabilities and a high-level programming language, APCOL, described in a companion paper. Two versions of APCOL exist: an instructional program implemented on the laboratory computer and a production version which resides on the IBM 360/67. The latter compiles object code that is brought to the laboratory machine over our high-speed line.

REAL-TIME CONCEPTS AND METHODS

There are obvious differences between real-time programming and the general acquisition of programming skills. The course is based on the assumption that all students have had some experience with a programming language. Hence, the basic concepts of sequential flow, iteration, assignment, and flow of control are assumed. We spend most of our class time on the special functions involved in interrupt processing and parallelism and on timing functions. These concepts, I believe, create the greatest difficulty for our students. In the laboratory, the students learn to program in APCOL using the on-line system (refer to Fig. 1). In this context, they come to grips with event mapping and device codes.

The principal instructional method of the course is a series of problem sets consisting of 10 preliminary problems and four additional sets of 3 problems each. Most of the concepts and most of the instructions available in the APCOL language are exercised by these problems. Students work individually at the main console of the DDP-116 surrounded by three of the special devices—a Teletype connected through a dataset to be used for output, a video monitor for alphanumeric displays, and a panel—the I/O panel—containing 16 illuminated pushbutton switches. The I/O panel is a bit-by-bit mapping of the general-purpose interface to the computer. These devices provide the necessary ingredients for most psychological experiments. The on-line APCOL compiler is loaded into the system and executed. A hard copy of the student's program and the output of any recorded data is immediately available on the ASR35 TTY.

The preliminary problem set makes it possible for the student to scan the list of APCOL instructions for key words in each problem and, with a little thought, to select one or more appropriate instructions. Each instruction can be executed immediately by typing “*!” at the end of the line. Feedback is immediate—what he types as an instruction and whatever is recorded as output on the Teletype, character strings and numbers on the video monitor, or illumination of specific indicator lamps on the I/O panel corresponding to the bits transmitted via the room interface.