A GENERAL EDUCATION COURSE: Mathematics, BASIC Programming, and Computer Literacy

Dr. Jerry W. Phillips
Professor of Mathematics and Education
Acting Dean of Academic Affairs
OAKLAND CITY COLLEGE
Oakland City, IN 47660

During a curriculum reorganization in a small midwestern liberal arts college, a faculty committee suggested that a general education course be developed which integrated mathematics and computers. This happened at the same time that an alumni group of former math teachers were asking about honoring a former professor. After meetings between the alumni, affected faculty members, and the Development Staff, a plan was formulated to memorialize the professor with a computer lab. The lab would be used for the general education course and other beginning courses in computing science. During this process, the college created both an associate degree in computing technology and a bachelors degree in applied math.

The course, Math 105, Mathematics and Computer Technology, was used as the vehicle to teach mathematics, BASIC computer programming, and computer literacy. It was also to be the hub of the new curriculum in computing science since limited resources made it impossible to offer a different course for applied Math/CS majors.

Two more factors influencing the development of the course were available hardware and student preparation. A network of TRS-80 model III and 4 computers has served well but has been limiting. Our student population has been such that several students have found that their lack of mathematical background and no computer experience has necessitated remedial courses before enrolling in Math 105. Remedial courses available are Developmental Mathematics, Beginning Algebra, and Introductory Programming. This approach has been very helpful for marginal and nontraditional students. We have found that for students with some computer literacy and average secondary mathematics backgrounds, the course is challenging but realistic.

In addition to the three primary objectives listed above there have been two supporting considerations during the course's development: 1) reducing math and/or computer anxiety, and 2) improving oral and written communication skills. The instructors have found that it is possible to develop the kind of caring atmosphere where the instructor personally interacts with the students and is able to reduce math and/or computer anxiety. Faculty offices are located next door to the lab which facilitates the answering of student questions. Also mathematics work-study students are available to assist students in the lab. Communication skills are encouraged by the careful evaluation of written assignments such as essays and computer programs. Students are expected to participate in class discussions and present homework solutions on the chalkboard. Students know they are expected to speak and write properly.

FIGURE 1

Relative Point Value of Course Requirements

Requirements	Possible Points	% of Total
3 Exams	300	46
3 Programming Projects	150	23
Miscellaneous Quizzes	100	15
5 Written Reports	50	8
Participation & Attendance	50	8
totals	650	100

Evaluation for the course is determined by the criteria listed in Figure 1 above. Ample opportunity is provided for students to garner points to achieve at a level appropriate to their ability. The three 100-point exams are approximately 40 points on mathematics concepts, approximately 30 points on tracing programs and supplying the output, and the remaining 30 points on flowcharting or coding mathematical algorithms. Usually ten 10-point quizzes are given during the semester to encourage attendance and completion of homework, to let students demonstrate mastery of the content, and to find areas of misunderstanding before an examination. The programming projects are assigned during the last half of the semester to give the students an opportunity to apply the programming concepts in a meaningful manner. These projects may be mathematical in nature, the entire class may do the same programming exercise, or students may choose to develop their individual applications. These projects are evaluated as to originality, style, completeness, and error-handling of inappropriate data. Flowcharts for each project are also considered in the evaluation.

The five written reports are usually assigned relatively early in the semester before students are working on the programming projects. This requirement of the course encourages students to read current articles related to computers and provides for evaluation of written communication. Students must identify at least one source with a bibliographic entry. Students are encouraged to relate the computer to their college major or explore topics such as the history of computing, particular software or hardware, technological limitations, and effects of computers on society. Students are encouraged to use a word-processor and write at least two pages. The final 50 points are available for the instructor to reward hardworking students for their diligent efforts or to penalize students who have not attended or put forth noticeable effort. This system of evaluation has been found reasonable in that marginal students who are willing to put forth considerable effort have been able to achieve a passing grade.

The content of the course closely follows the text, Using Computers in Mathematics by Elgarten, Posamentier, and Maresh. Mathematics areas covered include algebra, geometry, number theory,

probability, and statistics. A parallel development of BASIC programming concepts are introduced as the mathematics topics are being covered. See Figure 2 for specific topics of the two disciplines and their parallel presentation. (A deviation from standard outlining is used in this figure.)

The text was written as a first course in BASIC programming with the intent to teach BASIC programming and mathematics at the same time. The text is not hardware specific and is readily adaptable to common microcomputers. Although it does not deal with any storage procedures, after the first few weeks we include instruction on cassette tape storage due to its ease of use and availability. Students with diskette experience are permitted to use diskettes. Structured programming is emphasized throughout the course by encouraging a minimum number of GOTOs and illustrating modularity. Other good programming practices encouraged are the use of remarks, meaningful use of variables, user-friendly messages, proper error-handling, debugging practices, and efficiency considerations. These programming practices are also used in program evaluations. Students with previous programming experience often find this portion of the course the most frustrating because it requires them to break sloppy programming habits. Students have to be encouraged to use good programming practices and not to just get a program to run.

The course, Math 105, Mathematics and Computer Technology, appears to be meeting the needs of our liberal arts students. It seems to be teaching the fundamental mathematics, BASIC programming, and computer literacy foundations. The anxieties toward mathematics and/or computers seem to be helped by the informal, caring, "hands on" atmosphere of the class and laboratory environment. As we look to the future literacy needs of these liberal arts students, we especially want to increase the software packaging capability so that more instruction can be included on word-processing, data base, and spreadsheet packages.

The above description is merely a summary of the course developed for general education at our institution. If an instructor is interested in more background information, the overall computing science curriculum, student preparedness for the course, or a more in-depth analysis of the course, please see an article by Sermersheim from The Journal of Computing in Small Colleges. Mrs. Sermersheim has been one of the instructors for this general education course.

Bibliography

Elgarten, G.H., Posamentier, and Moresh, Using Computers in Mathematics, Addison-Wesley, 1983.

Sermersheim, Robin A., "Killing Three Birds with one General Education Requirement," The Journal of Computing in Small Colleges, Consortium for Computing in Small Colleges, P.O. Box 329, Evansville, Indiana, Volume 3, Number 2, November 1987.

FIGURE 2

MATH 105 COURSE OUTLINE

NOTE: Numbering Code is as follows:
Roman Numerals indicate title of unit.
Capital letters indicate mathematical topics.
Hindu-Arabic digits indicate programming concepts.

I. INTRODUCTORY PROGRAMMING CONCEPTS
- 1. Computer Programming and Operations
- A. Solving Simple Equations
  - 2. Preparing a Problem for Computer Solution
  - 3. Variables
  - 4. Order of Operations
  - 5. Assignment Statements
  - 6. INPUT, PRINT, and END instructions
  - 7. Tracing Programs
  - 8. Entering and RUNning a program
  - 9. Formatting Output
- B. Perimeter and Area Calculations
  - 10. Initialization of Variables
  - 11. EDITing Programs
  - 12. Flowcharting Algorithms
  - 13. Unconditional Branching (GOTO)
  - 14. Conditional Branching (IF-THEN)

II. ELEMENTARY ALGEBRA CONCEPTS
- A. Fahrenheit and Celsius Conversions
  - 1. User Prompts with INPUT
  - 2. Statement Separators
- B. Divisibility
  - 3. Greatest Integer Function
- C. Absolute Value
  - 4. Absolute Value Function
  - 5. TAB Function
- D. Newton-Raphson Method for Approximating Square Roots
  - 6. Truncation
  - 7. Rounding
  - 8. Square Root Function
- E. Solving Systems of Linear Equations by Reduction
- F. Determinants
- G. Cramer's Rule for Solving Systems of Linear Equations
  - 9. Avoiding Endless Loops
  - 10. Use of Menus
- H. Galileo's Gravitational Formula
  - 11. Use of a Counter
- I. Invoicing Problems
  - 12. READ-DATA Statements
  - 13. Use of an Accumulator
- J. Finding the Arithmetic Mean
  - 14. FOR-NEXT Loop
- K. Factorials

III. GEOMETRY CONCEPTS
- A. Triangle Inequality Problem
  - 1. Logical Operators AND and OR
- B. Pythagorean Theorem
  - 2. String Variables
  - 3. Nested Loops
- C. Classification of Triangles
  - 4. Exchanging Contents of Two Variables
- D. Heron's Formula for Area of a Triangle
- E. Distance Formula in Two and Three Dimensions
- F. Congruency
- * G. Circle Intersection Problem

IV. OTHER ALGEBRAIC TOPICS
- A. Solving Quadratic Equations
- B. Classifying the Roots of a Quadratic Equation
- * C. Euler's Method to Solve Diophantine Equations
- * D. Arithmetic Sequences and Their Sums
- * E. Harmonic Sequences and Means

V. NUMBER THEORY
- A. Prime and Composite Numbers
- B. Euclid's Division Algorithm
- C. Perfect, Abundant, and Deficient Numbers
- D. Amicable Numbers
  - 1. Subroutines
- E. Finding the Greatest Common Divisor
- F. Finding the Least Common Multiple
- G. Operations with Fractions
  - 2. One-dimensional Arrays
  - 3. Replacement Sort
  - 4. Bubble Sort
  - * 5. String functions (MID$ and LEN)

VI. PROBABILITY AND STATISTICS
- A. Fundamental Counting Principle
- B. Permutations
- C. Combinations
- D. An Introduction to Probability
- E. The Birthday Problem
- F. Using Random Numbers
  - 1. The Random Number Generator Function
- G. Measures of Central Tendency: Mean, Median, Mode
- H. Data Frequency Charts
- I. Summation Notation
- J. Measuring Dispersion: Variance and Standard Deviation

* Optional Topics