

SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: BASIC PROGRAMMING

Code No.: CET 100

Program: CIVIL & ARCHITECTURAL

Semester: ONE

Date: SEPTEMBER 1987

Author: K. CHENG

New: \_\_\_\_\_ Revision: xx

APPROVED: *L.P. Chazitt*  
Chairperson

87-08-17  
Date

CALENDAR DESCRIPTION

BASIC PROGRAMMING

CET 100

---

Course Name

---

Course Number

**PHILOSOPHY/GOALS:**

This course is an introductory computer programming course for the Architectural & Civil (based on the course for first semester Electrical, Electronic, and Instrumentation Technician students). Because it is a first semester course, it is not intended to be a rigorous problem solving course, but, rather, a course that will introduce the student to the BASIC language and provide the student with the necessary skills in using the computer required for successive courses.

**METHOD OF ASSESSMENT (GRADING METHOD):**

SEE ATTACHED SHEET ENTITLED **GRADE REQUIREMENTS.**

**TEXTBOOK:**

**IBM PC: AN INTRODUCTION TO THE OPERATING SYSTEM, BASIC PROGRAMMING AND APPLICATIONS**, by L.J. Goldstein, 3rd. Ed., Pentice Hall.

## GRADE REQUIREMENTS

### CET 100

#### BASIC PROGRAMMING

#### (ARCH/CIVIL)

Your final grade in CET 100 will be determined on the basis of three tests to be administered during the semester. Each will examine your knowledge of a number of topics and will be administered within a week of completing those topics. The topics covered by each of the four tests are as follows:

TEST 1 -- TOPIC NUMBER I  
          TOPIC NUMBER II

TEST 2 -- TOPIC NUMBER III

TEST 3 -- TOPIC NUMBER IV  
          TOPIC NUMBER V

The 3 tests are of equal weight (i.e. each of the tests is worth 30% of your final grade and 10% is miscellaneous). In order to obtain your letter grade, the following percentage/letter grade equivalents will be used:

A = 80 - 100%

B = 69 - 79%

C = 55 - 69%

R = 0 - 54%

Should you find yourself with an R grade at the end of the semester, in order to upgrade your mark to C, you will be required to pass a make-up examination covering the entire course content.

Prior to administering any test, you will be notified a full week in advance. Should you for any reason not be able to be in attendance on a day for which a test has been scheduled, it is your responsibility to notify the instructor prior to the test. If your reasons are acceptable, a date will be set during which you may write the test you have missed.

**COURSE OUTLINE**  
**CET 100**  
**BASIC PROGRAMMING**

| TOPIC NUMBER | TOPIC DESCRIPTION   |
|--------------|---|
| <b>I</b>     | <b>INTRODUCTION</b> <ul style="list-style-type: none"><li>- interaction with the computer</li><li>- the PRINT statement<ul style="list-style-type: none"><li>- the comma</li><li>- the semi-colon</li></ul></li></ul>   |
| <b>II</b>    | <b>WRITING SIMPLE PROGRAMS</b> <ul style="list-style-type: none"><li>- program format<ul style="list-style-type: none"><li>- the END statement</li><li>- line numbers</li></ul></li><li>- BASIC commands<ul style="list-style-type: none"><li>- the RUN command</li><li>- the LIST command</li><li>- the NEW command</li><li>- the SAVE command</li><li>- the SCRATCH command</li></ul></li><li>- providing program data<ul style="list-style-type: none"><li>- the ASSIGNMENT statement</li><li>- VARIABLE names</li><li>- the INPUT statement</li><li>- the READ and DATA statements</li></ul></li><li>- documentation<ul style="list-style-type: none"><li>- the PRINT statement</li><li>- the REM statement</li></ul></li></ul> |
| <b>III</b>   | <b>NUMBERS, VARIABLES &amp; OPERATIONS</b> <ul style="list-style-type: none"><li>- large and small numbers</li><li>- string variables</li><li>- integers</li><li>- arithmetic operations</li><li>- exponentiation</li><li>- square root</li><li>- complicated expressions</li><li>- the SIN, COS, TAN, &amp; ATN functions</li></ul>  |
| <b>IV</b>    | <b>PROGRAM CONTROL</b> <ul style="list-style-type: none"><li>- the GOTO statement</li><li>- the IF...THEN statement</li></ul>   |

V

**LOOPS AND DECISIONS**

- the FOR/NEXT statement
  - simple loops
- nested loops

VI

**SUBSCRIPTS AND ARRAYS**

- subscripts
- the DIM statement
- one-dimensional arrays
- the RESTORE statement
- two-dimensional arrays (matrices)

VII

**FORMATTING**

- the TAB function