

CE331-5
CODE NO.

HIGH LEVEL LANGUAGE PROGRAMMING
COURSE NAME

SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: HIGH LEVEL LANGUAGE PROGRAMMING

Code No.: CET331-5 Semester: 5

Program: ELECTRICAL/ELECTRONIC TECHNOLOGY

Author: Mark Allemang

Date: SEPT., 1994 Previous Outline Dated: SEPT., 1993

APPROVED: *L.P. Crockett* 94-09-06
Dean Date



HIGH LEVEL LANGUAGE PROGRAMMING
COURSE NAME

CET331-5
CODE NO.

TOTAL CREDITS : 5

PREREQUISITES: CET228-5 MICROPROCESSOR CIRCUITS AND
 APPLICATIONS

I. PHILOSOPHY/GOALS:

This course introduces the student to the C programming language. It is the students first exposure to a "high level" programming language. It is not an introduction to programming concepts and entry to this course requires that the student has successfully completed CET228-5.

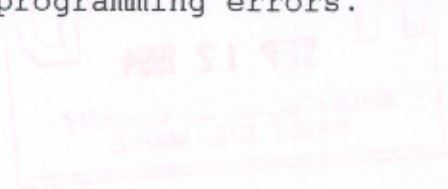
The student is introduced to C in two environments including VAX C and PC based Turbo C. This course will also serve as an introduction to VAX/VMS with the student learning the commands and utilities required to develop programs within the VMS environment.

Practical skills will be developed with a series of C programming assignments done in both a PC and a VAX/VMS environment.

II. STUDENT PERFORMANCE OBJECTIVES:

Upon successful completion of this course the student will be able to:

1. Logon to a computer within a typical computer network.
2. Utilize EMAIL to send and receive C source files.
3. Describe the program development sequence, the products and the potential errors at each stage.
4. Develop C programs in a PC-DOS and VAX/VMS environment.
5. Write programs that utilize the following features of C:
 - the various data and variable types
 - arithmetic and assignment operators
 - input/output statements
 - control statements and relational operators
 - bitwise operators
 - microprocessor input/output statements for direct hardware control
 - arrays and character strings
 - pointer variables
 - functions and arguments
6. Describe a solution to a programming problem using Pseudocode.
7. Employ a High Level Language Debugger to help detect typical programming errors.



III. TOPICS TO BE COVERED:

1. Introduction to the Sault College Computer Network and the VAX/VMS Environment
2. The Program Development Sequence
3. Introduction to C / VAX C, Data and Variable types, Operators, Expressions and Statements
4. Formatted Input/Output
5. C Control Statements and relational operators.
6. Introduction to Turbo C and its Debugging Features
7. Pseudocode
8. The Bitwise operators and Controlling The Hardware from C
9. Arrays, Strings and Pointers
10. Functions

IV. LEARNING ACTIVITIES/REQUIRED RESOURCES

1. Introduction to the Sault College Computer Network and the VAX/VMS Environment

Learning Activities

- listen to a presentation about the college computer network facilities
- use these facilities to Log-on and off the Novell file server and the VAX computer system
- Discuss the concept of files and directories and the method of naming files on the VAX
- listen to a presentation on DCL (Digital Command Language) to:
 - a. Display the contents of a file on the system.
 - b. Delete files.
 - c. Purge files.
 - d. Rename files.
 - e. Create and maintain directory structures and their associated files.
- complete exercises using the above DCL functions
- listen to a presentation on the EDT editor
- utilize the EDT Editor to create text files and write example C programs
- listen to a presentation on VAX EMAIL
- utilize EMAIL to send text files to other users.

Resources:

overheads

=====
2. The Program Development Sequence

Learning Activities:

- listen to a presentation on the process of editing, compiling, linking and running a program and discuss the products and potential errors at each stage
- discuss the difference between Source, Object, List and Executable files.
- Edit, compile, link and run an example C program provided by the instructor.

Resources:

Text ch 1, overheads

=====
3. Introduction to C / VAX C, Data and Variable Types, Operators, Expressions and Statements

Learning Activities:

- listen to a presentation introducing the C language
- Write simple C programs to type a message on the screen. Compile link and run the programs.
- listen to a presentation on the types of constants and variables and their appropriate use in C programs
- write programs to print out the contents of different types of variables on the screen
- listen to a presentation on the C mathematical operators and their precedence
- write programs that make use of the various mathematical operators and show examples of precedence

Resources:

Text ch 2,3,5 , overheads

=====
4. Formatted Input/Output

Learning Activities:

- listen to a presentation on the use of the I/O functions printf and scanf and the use of the I/O Conversion Specifiers
- write programs that provide examples of the use of printf, scanf and the various conversion specifiers

Resources:

Text ch 4, overheads

- =====
- 5. C Control Statements and relational operators.
 - 6. Introduction to Turbo C and its Debugging Features

Learning Activities:

- discuss the concept of program repetition/looping
- listen to a presentation on the use of the various control statements and relational operators
- write programs that make use of the various control statements.
- write programs that make complex decisions by using relational operators
- listen to a presentation on Turbo C and utilizing its debugging features
- develop programs using Turbo C and utilize its Debugging features detect errors in programs

Resources:

Text ch 1,6,7 , overheads

- =====
- 7. Pseudocode

Learning Activities:

- listen to a presentation on the use of Pseudocode as a tool for developing complex programs
- write a solution to a programming problem using pseudocode

Resources:

Text pg 162 , overheads

- =====
- 8. The Bitwise operators and Controlling The Hardware from C

Learning Activities:

- listen to a presentation on using of the C Bitwise operators
- listen to a presentation about controlling the IBM-PC Hardware from a C program
- write programs that control the IBM-PC speaker directly from C

Resources:

overheads

HIGH LEVEL LANGUAGE PROGRAMMING
COURSE NAME

CET331-5
CODE NO.

=====

9. Arrays and Strings and Pointers

Learning Activities:

- listen to a presentation on the concept of an Array variable and how to declare and initialize an array
- discuss the characteristics of a character string in C
- write programs which utilize string functions to manipulate and test strings
- listen to a presentation on the pointer type variable and how pointers are used in C
- write programs that utilize pointers instead of array subscripts

Resources:

Text ch 10 , overheads

=====

10. Functions

Learning Activities:

- Discuss "program modularity" through the use of functions.
- listen to a presentation on the rules for using functions in C.
- write programs that make use of functions.

Resources:

Text ch 9 , overheads

=====

V. METHOD OF EVALUATION:

2 THEORY TESTS	60%
ASSIGNMENTS and LAB WORK	35%
QUIZZES	5%

(The percentages shown above may vary where circumstances warrant.)

- Notes:
1. Lab work and assignments must be complete to the instructor's satisfaction for a passing grade to be achieved.
 2. Before tests the instructor will provide details of the specific objectives to be tested.

HIGH LEVEL LANGUAGE PROGRAMMING
COURSE NAME

CET331-5
CODE NO.

GRADING SCHEME

A+	90	-	100%
A	80	-	89%
B	70	-	79%
C	55	-	69%
I	Incomplete		
R	Repeat		

UPGRADING OF INCOMPLETES

When a student's course work is incomplete or final grade is below 55%, there is the possibility of upgrading to a pass when a student meets the following criteria:

1. The student's attendance has been satisfactory.
2. An overall average of at least 40% has been achieved.
3. The student has not had a failing grade in all of the theory tests taken.
4. The student has made reasonable efforts to participate in class and complete assignments.

ASSIGNMENTS AND LAB ACTIVITIES:

Lab activities and assignments represent a very important component of this course in which practical 'hands-on' skills will be developed. Because of this, lab attendance is mandatory and the satisfactory completion of all assignments is required. It is the student's responsibility to discuss absences from regularly scheduled labs with the instructor so that alternate arrangements (where possible) can be made to complete the lab requirements.

A penalty for late assignments will be applied unless there are extenuating circumstances. A 10% per week penalty will be applied. After 4 weeks late assignments will not be accepted for credit.

It is acceptable that students consult with each other in relation to their assigned problems. However, it is unacceptable to copy programs written by someone else and submit them as your own work. Where plagiarism or copying is found and it is impossible to determine whose original work it is, a mark of zero will be assigned to all assignments involved.

ATTENDANCE:

Absenteeism will affect a student's ability to succeed in this course. Absences due to medical or other unavoidable circumstances should be discussed with the instructor, so that remedial activities

HIGH LEVEL LANGUAGE PROGRAMMING
COURSE NAME

CET331-5
CODE NO.

can be scheduled. A Quiz or Test missed because of an unauthorized absence will result in a zero grade being assigned.

VI. REQUIRED STUDENT RESOURCES:

TEXT BOOKS:

1. The Waite Group's New C Primer Plus (Second Edition) by Mitchell Waite & Stephen Prata.

DISKETTES:

Two 3-1/2" HD Diskettes.

VII. SPECIAL NOTES:

1. Students with special needs (eg. physical limitations, visual or hearing impairments, or learning disabilities) are encouraged to discuss any required accommodations confidentially with the instructor.
2. Your instructor reserves the right to modify the course as deemed necessary to meet the needs of students or take advantage of new or different learning opportunities.
3. The learning topics and activities will not necessarily be covered in the order shown in this course outline.