Introduction to Computing

School of Technology and Trades

Course Number: CST8110	**Contribution to Program:** - Computer Science - Core - Computer Engineering Technology - Core - Computer Systems Technician - Core	**Educator(s):** Arnold Betz Gary Mroz
Applicable Program(s): Computing Science Technology Computer Engineering Technology Computer Systems Technician	**AAL:** 01	**Approval Date:** 1995/11/20 **Approved By:** David Fisher Title: Chairperson, Computer Studies
Course Hours: Delivered: 64 Normative: 64	**Prerequisites:** nil **Corequisites:** nil	**Approved for Academic Year:** 1997 - 1998

COURSE DESCRIPTION

Students in this course learn fundamental problem solving methodologies needed in computing, such as structured analysis, structured design, structured programming and modular programming. Pseudocode, flowcharts and the C programming language are used to develop solutions to problems of increasing complexity. Students learn the basics of computer programming through the C programming language, with emphasis on structured programming, programming style and documentation to achieve program reliability and maintainability. The course reinforces theory with applications by means of practical laboratory assignments.

RELATIONSHIP TO PROGRAM LEARNING OUTCOMES	
This is a vocational course that supports the following vocational program standards:	**This course contributes to your program by helping you to achieve the following provincial generic skills standards:**
problem specification and analysis problem solving methodologies for computing programming and documentation style structured C language programming program testing and debugging methodologies	basic computer literacy workplace discipline

COURSE CURRICULUM

I. Course Learning Requirements/Embedded Knowledge and Skills

When you have earned credit for this course, you will have demonstrated ability to:

- Use fundamental concepts of computer programming to solve problems using computers.

- Use the basic thought processes needed in computing and programming.

- Express solutions in pseudocode, flowcharts and C language.

- Interpret solutions that have been presented in flow chart form.

- Create, test, debug and document structured, modularized C programs.

- Understand and describe how a computer operates, particularly the methods of storage of different data types, along with the implications of those types.

- Exercise self-control with respect to attendance, punctuality, working to deadline and respect for other persons in the workplace.

II. Learning Resources

The course consists of two hours of lectures and two hours of laboratories per week.

Laboratory assignments will be done in laboratories equipped with personal computers, the MS-DOS operating system, and the C programming language. Professors will be in the laboratory to provide assistance. Laboratory assignments will be closely integrated with the lecture material and will normally exercise all previously learned knowledge and skills. Students should expect to have to do work outside scheduled laboratory hours to complete laboratory assignments on time.

Textbooks:
1. Problem Solving Methodology and Programming Style, Computer Studies Department

2. C How To Program, 2nd Edition, Deitel & Deitel

References:
1. C By Example, Perry

2. Jamsa's 1001 C/C++ Tips, Jamsa

III. Teaching/Learning Methods

Lecture sessions will present the theoretical material of the course, aided by use of overhead projections, computer demonstrations and brief lecture notes. Students will be expected to find and read applicable material in the textbook, and to be prepared to answer oral or written questions in following lectures. Students are encouraged to ask questions during lectures and to consult with the professors on topics which they do not clearly understand.

Laboratory sessions will provide opportunities for students to apply the lecture material to a series of assignments which are closely integrated with the current lecture materials, require the use of previously-learned methods, and are increasingly complex as the course progresses. Students should seek advice and help from the professors in the laboratory.

IV. Learning Activities and Assessment

Lectures will normally include examples of the application of the methods being taught. These examples may be worked out on the blackboard, or may be presented by means of transparencies on an overhead projector, or may be projections of actual computer screen outputs. Students are expected to observe carefully and to ask for clarifications or further examples.

Topics covered include:

1. Computer number theory. (2 weeks)

    Representation of integers in binary and hexadecimal form, negative integers using two's complement, floating-point numbers, and characters (the ASCII character set).

2. Problem solving principles and algorithm design. (4 weeks)

    The input-process-output model of computing. The elements of pseudocode: sequence, selection and iteration. Development of structured pseudocode using stepwise refinement and flowcharting techniques.

3. Elements of computer programming. (4 weeks)

    Data types (char, int, long, unsigned char, unsigned int, unsigned long, float, double) and declarations, input, output, assignment, calculation, use of library functions.

4. C language control structures. (3 weeks)

    (*if...else*, *while*, *do...while* and *for*). Combining and nesting the constructs.

5. C program modularity. (2 weeks)

    Writing and using C-language functions involving parameters passed by value.

Laboratories will provide opportunities for hands-on use of the computer to write, test and debug computer programs, with the professor in attendance and on call for assistance. Students are expected to work on their own if possible, to work with other students in developing solutions, and to ask for assistance from the professor when necessary. Students may be required to show completed pre-lab portions of assignments before being admitted to the laboratory.

Assessment of student learning will be done by means of class tests and quizzes, marked assignments and a written final examination.

V. Evaluation/Earning Credit

The following will provide evidence of your learning achievement:

Assessment of student learning will be done by means of class tests and quizzes and a written final examination, all based on the material covered in lectures, together with an assessment of the student's performance in laboratory assignments.

Laboratory attendance is compulsory, and absence from three or more laboratory sessions without the consent of the instructor will result in a final grade of F.

All laboratory assignments must be successfully completed in order to receive credit. Assignments will be penalized for lateness: late less than 1 week, -20%; late 1 week or more, -100%.

Plagiarism (work submitted by the student which is substantially the work of other persons) will not be tolerated. Students who knowingly allow their work to be plagiarized will receive the same sanctions as the plagiarizer. The first occurrence of plagiarism will result in a mark of zero for the assignment and the students will be required to redo it. A second occurrence will result in a grade of F.

The three factors in the final grade are:

1. Class Tests and Quizzes (35% of total marks)
2. Final examination (40% of total marks)
3. Laboratory Assignments (25% of total marks)

The final grade will be calculated as the total of the three factors. Laboratory assignments will not be included in the final grade unless the student achieves at least 38 out of 75 on the total of class tests, quizzes and final examination.

VI. Prior Learning Assessment

Evidence of learning achievement for PLA candidates will include:

Completion of a challenge test with a breadth of coverage and level of difficulty equivalent to the final examination in the course, with a grade of at least C, and at the assessor's discretion, successful completion of a computer programming assignment comparable to the final assignment in the course.

RELATED INFORMATION

If you are a student with a disability please identify your needs to the professor and/or the Centre for Students with Disabilities (CSD) so that support services can be arranged for you. You can do this by making an appointment at the CSD, Room C142, Ext. 7683 or arranging a personal interview with the professor to discuss your needs.

Students, it is your responsibility to retain course outlines for possible future use to support applications for transfer of credit to other educational institutions.