

**NOVA COLLEGE-WIDE COURSE CONTENT SUMMARY
CSC 130 - SCIENTIFIC PROGRAMMING (3 CR.)**

Course Description

Introduces a science-oriented, high level programming language. Studies the language and its application in problem solving. Lecture 3 hours per week.

General Course Purpose

This course will introduce problem solving methods and algorithm development, teach a high level programming language, and teach how to design, code, debug, and document programs using techniques of good programming style. A high level programming language will be used. This course prepares students for Computer Science I.

Course Prerequisites/Corequisites

Readiness for ENG 111, MTH 161 or MTH 167, or equivalent.

Course Objectives

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

- Analyze a simple problem and develop an algorithm for its solution using various techniques, to include top-down development, pseudo code, flowcharts, hand-checking, well-chosen test data, and stubs and drivers.
- Implement an algorithm in a high-level computer language, demonstrating good style and appropriate documentation, using various language features to include statements, data types, operations, input and output, simple control structures, subprograms, and parameter passing.
- Distinguish among typical system software and application software, high-level and low-level computer languages, as well as computer language compilers and interpreters.
- Distinguish among major components of computer hardware.

Major Topics to be Included

- I. Overview of the computer system
 - A. Hardware and software
 - B. Systems and applications software
- II. Problem analysis and algorithm development
 - A. Top-down program development
 - B. Algorithm representation in pseudocode and flowcharts
 - C. Algorithm verification by hand-checking
 - D. Choosing test data
 - E. Stubs and drivers
 - F. Programming style and documentation
- III. Programming language
 - A. Statements
 - B. Data types
 1. Numeric and Character Types
 2. One- and Two-Dimensional Arrays
 - C. Operations

1. Arithmetic
 2. Logical
 3. Character
- D. Input/output
- E. Control structures
1. Sequential
 2. Selection
 3. Repetition
- F. Subprograms and parameters
1. User-defined and library subprograms
 2. Formal vs. actual parameters
 3. Calls by value and by reference
 4. Scope of variables
- IV. **Other** optional content
- A. Overview of object-oriented programming
- B. Scientific software packages
1. Using a package such as Mathematica, Maple, Derive, Minitab, SAS, Excel, Octave, R, etc. for problem solving
 2. Writing programs using the programming language built into such a software package
- C. Suggested Scientific Problems to Solve
1. Fundamental Statistics
 2. Summation of Terms of a Series
 3. Approximation Methods
 - a. Newton's Method
 - b. Numerical Integration
 4. Systems of Equations
 5. Finding Least Common Multiples, Greatest Common Divisors
 6. Matrix Operations

Suggested Time Allocation per Topic

In order to standardize the core topics of CSC 130 so that a course taught at one campus resembles the same course taught at another campus, the following student-contact-hour minima per topic have been adopted. Of course, the topics cannot be followed sequentially. Many topics are taught best as an integrated whole, often revisiting the topic several times, each time at a higher level. There are normally 48 student-contact-hours per semester for a three-unit course. The last category, **other** optional content, leaves ample time for an instructor to tailor the course to special needs or resources.

Ref	Topic	Hours	Percent
I	Overview of the computer system	2	4
II	Problem analysis and algorithm development	12	25
III	Programming language	24	50
IV	Other optional content	6	13
	Exams and Quizzes	4	8
	Total	48	100