

**NOVA COLLEGE-WIDE COURSE CONTENT SUMMARY  
CSC 206 – ASSEMBLY LANGUAGE (3 CR.)**

**Course Description**

Examines assembly language programming. Includes the use of macros, linkers, loaders, assemblers and interfacing assembly language with hardware components. Lecture 3 hours per week.

**General Course Purpose**

The extent to which each topic is discussed and the ordering of topics depends on facilities available. Enough assembly language details should be covered and programming projects assigned so that the student gains experience in programming in assembly language for a specific computer. However, concepts and techniques that apply for a broad range of computers should be emphasized.

**Course Prerequisites/Corequisites**

Prerequisite: CSC 202

**Course Objectives**

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

- Convert text into ASCII and vice versa
- Represent signed decimal integers in binary, two's complement notation, and vice versa
- Write binary numbers in hexadecimal notation, and vice versa
- State why and how a symbol table is used in the assembly process
- Write a simple macro
- Explain the purpose of a macro and how a macro is expanded in the assembly process
- State the purpose of a linker and a loader, the use of libraries, as well as the reasons for and the implementation by the linker of relocatable object code
- Write, assemble, link, debug, and run an assembly language program that uses the following: different addressing modes, arithmetic operations, logical operations, iteration, selection, subroutines and parameter passing, character string operations, arrays, and system calls
- State the mechanism of interrupt-driven I/O and the use of I/O ports
- Write an assembly language subroutine that requires parameters and call it from a high-level language

**Major Topics To Be Included**

- I. Introduction and Review
  - A. Data representation
    1. Numbers
    2. Text
  - B. Computer Organization
    1. Virtual Machine concept
    2. Specifics of Machine Used in the Course
- II. Assembly Language Fundamentals
  - A. Addressing
    1. Register
    2. Immediate
    3. Memory/Direct
    4. Indirect
    5. Indexed
    6. Absolute vs. Relative
  - B. Operations

1. Arithmetic
  2. Logical
  3. Data Transfer
- III. Programming
- A. Control Structures
    1. Flags
    2. Selection/Branching
    3. Iteration
  - B. Subroutines
    1. Calling Methods
    2. Parameters and Passing Methods
    3. Recursive Subroutines
  - C. Character String Operations
  - D. Data Structure Implementation
    1. Array
    2. Record/Structure
    3. Pointer
    4. Set
  - E. Input/Output
    1. Memory Mapped
    2. Direct Memory Access
  - F. System Calls
- IV. The Assembly Process
- A. Directives
  - B. Symbol Tables
  - C. Linkers and Loaders
    1. Object Module
    2. Libraries
    3. Relocation
  - D. Macros
    1. Definition, Call and Expansion
    2. With Parameters
    3. Conditional
    4. Implementation
- V. Interface of Assembly Language with Hardware
- A. Interrupts
  - B. Ports
- VI. High-Level Language Interface
- A. Stack Frame
  - B. Parameters
  - C. Linking
- VII. Other Topics (optional)
- A. Other Assembly Languages
    1. Single Register (Accumulator) Systems
    2. General Register Systems
    3. Stack Machines
    4. RISC versus CISC
  - B. Operating System Interface
    1. Console Programming
    2. Graphical Programming
    3. Device Drivers
    4. Interrupt and Exception Handling
  - C. Advance Memory Management
    1. Virtual Memory
    2. Multitasking
  - D. Instruction Encoding

### Suggested Time Allocation per Topic

In order to standardize the core topics of CSC 206 so that a course taught at one campus is equivalent to the same course taught at another campus, the following student-contact-hours per topic are recommended. There are normally 48 student-contact-hours per semester for a three-unit course. The last category, *Other/Enhance*, leaves ample time for an instructor to tailor the course to special needs or resources.

Ref	Topic	Hours	Percent
I	Introduction / Review	3	6
II	Assembly Language Fundamentals	--	--
A	Addressing	3	6
B	Operations	3	6
III	Programming	--	--
A	Control Structures	2	4
B	Subroutines	3	6
C	Character String Operations	3	6
D	Data Structure Implementation	2	4
E	Input/Output	3	6
F	System Calls	2	4
IV	Assembly Process	--	--
A	Directives	2	4
B	Symbol Tables	2	4
C	Linkers and Loaders	2	4
D	Macros	2	4
V	Interface of Assembly Language with Hardware	2	4
VI	High-Level Language Interface	2	4
VII	<i>Other</i> optional content or <i>enhance</i> the above	8	19
	Exams and Quizzes	4	9
	Total	48	100