**Course Contents**

Introduction to Assembly Language:

Computer Programming Languages (machine, low level, high level language), Why am I learning Assembly Language?, What background should I have?, What is an assembler?, What hardware/software do I need?, What types of programs will I create?

Basic Concepts:

Virtual Machine Concept: Specific Machine Levels, High level language, Assembly Language, Instruction Set Architecture (ISA), Digital Logic

Data Representation: Binary Number System, Hexadecimal Number System, Signed Representation, Character Storage

Boolean Operations

Boolean algebra(Not, AND, OR, XOR),Operator Precedence, Truth Tables

Processor Architecture:

* + General Concepts
    - Basic microcomputer design
    - Instruction execution cycle
    - Reading from memory
    - How programs run
  + IA-32 Processor Architecture
    - Instruction Execution Cycle
    - Cache Memory
    - Load and execute process
    - Multitasking
    - Modes of operation
  + Basic execution environment
    - Addressable memory
    - General-purpose registers
    - Index and base registers
    - Specialized register uses
    - Status flags
    - Floating-point, MMX, XMM registers
  + Intel Microprocessor history
  + CISC and RISC
  + IA-32 Memory Management
    - Real-address mode
    - Calculating linear addresses
    - Protected mode
    - Multi-segment model
    - Paging

Assembly Language Fundamentals

* Basic Elements of Assembly Language
* Example: Adding and Subtracting Integers
* Assembling, Linking, and Running Programs
* Defining Data
* Symbolic Constants
* Real-Address Mode Programming

*Lab Work:*

*Introduction to assembly language tools: (Assembler, Linker, Debugger, Editor)*

*Installing MASM 6.15*

*Displaying a welcome statement (assembling, linking, and running the program)*

*Discussion about program structure*

*Homework Exercise*

Data Transfers, Addressing, and Arithmetic

* + Data Transfer Instructions
    - MOV, XCHG
    - Operand types (Immediate, register, memory)
  + Addition & Subtraction (INC, DEC, ADD, SUB, NEG)
  + Flags Affected by Arithmetic

*Lab Work:*

*Taking input from user*

*Addition and subtraction of numbers*

*Displaying flags*

*Homework Exercise*

JMP and Loop Instructions

* JMP Instruction
* LOOP Instruction
* LOOP Example
* Nested Loop
* Summing an Integer Array
* Copying a String

*Lab Work:*

*Implementation of looping structure*

*Homework Exercise*

Conditional Processing

* + Boolean and Comparison Instructions
    - CPU Status Flags
    - AND Instruction
    - OR Instruction
    - XOR Instruction
    - NOT Instruction
    - Applications
    - TEST Instruction
    - CMP Instruction

*Lab Work:*

*Implementation of comparison & Boolean instructions*

*Homework Exercise*

* + Conditional Jumps
    - Based on: Specific flags, Equality, Unsigned comparison, Signed Comparisons
    - Conditional Loop Instructions (While Loop)
    - Conditional Structures (translation from high level language to Assembly language)

*Lab Work:*

*Implementation of conditional structures*

*Homework Exercise*

Procedures

* + Library Procedures - Overview
  + Stack Operations
    - PUSH, POP
    - PUSHA, POPA (for 16-bit general purpose registers)
    - PUSHAD, POPAD (for 32-bit general purpose registers)
    - PUSHFD, POPFD (EFLAG register)
  + Defining and Using Procedures
    - CALL and RET Instructions
    - Nested procedure calls
  + Program Design Using Procedures

*Lab Work:*

*Implementation of stack operations*

*Implementation of procedures*

*Homework Exercise*

Integer Arithmetic

* + MUL, IMUL
  + DIV, IDIV
  + CBW, CWD, CDQ Instructions

*Lab Work:*

*Implementation of arithmetic operations*

*Discussion about mini project*