

COURSE DESCRIPTION

Department and Course Number	CS 340	Course Coordinator	Vladimir Akis
Course Title	Assembly Language and Systems Programming	Total Credits	4

Current Catalog Description:

Assembly language; addressing techniques; subroutines; macros; system input/output; interrupts and traps; assemblers; linkers; loaders; microprocessors.

Textbook:

Jones, W. B. *“Assembly Language for the IBM PC Family”*, Scott Jones Publishers, 1996.

References:

Giles, W. B. *“Assembly Language Programming for the Intel Family”*, 2nd Edition, MC Graw Hill.

Course Goals:

At the end of the course, students are able to

- Write Assembly Language code in stand-alone programs and interfaced with high level languages.
- Understand low level programming, the two pass assembly process, and the Fetching Cycle.
- Use interrupts and understand their mechanism.
- Know basic computer architecture and computer hardware.
- Files at the Assembly Language level.

These course goals contribute to the success of **Student Learning Outcomes 1, 2, 3, 5, and 6**.

Prerequisites by Topic:

Using Operating Systems and Networks for Programmers.

Major Topics Covered in the Course:

- Assembly language instructions
- The Assembler and two pass assembly.
- Addressing
- Procedures, Library and Macros
- Assembler Directives
- Interface with high level languages

- File handling

Laboratory Projects (specify number of weeks on each):

Several (5 or more) programming projects in assembly language covering all the major topics. At least one project every two weeks.

Estimate Curriculum Category Content (Quarter Hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms			Data Structures		
Software Design		1.0	Prog. Languages		1.0
Comp. Arch.		2.0			

Oral and Written Communications:

Written documentation of software built in labs and homework assignments.

Social and Ethical Issues:

No significant component.

Theoretical Content:

No significant component.

Problem Analysis:

In the first part of the course, students learn the fundamentals of computer organization, a basic assembly language instruction set, and addressing modes. Next they are introduced to procedures and macros. They learn to write stand-alone assembly language programs, implementing the standard assembly directives. In the later part of the course, students learn how to use system interfaces

Solution Design:

Solution design in this course involves the use of Assembly Language appropriate for certain programming tasks requiring intensive hardware access and for developing efficient large programs.