

***Preliminary***  
**Master Course Syllabus**  
School of Engineering and Computer Science  
Washington State University Vancouver

**CS 261**  
**C and Assembly Language Programming**  
3 Semester Hours

**Catalog Description**

C language concepts, professional practices and C programming, module linkage; assembly language concepts and programming.

**Prerequisite Courses**

CS 260 – Computer Organization

**Prerequisite Topics**

- Proficiency using an imperative programming language
- Knowledge of computer instruction set architecture
- Experience with computer data representation
- Knowledge of computer memory and I/O architecture

**Measured Course Outcomes**

Students taking this course will:

1. Design, code and debug programs in the C programming language, implementing dynamic, self-referential data structures.
2. Make effective use of a debugger to find errors in programs.
3. Design, code and debug a program, using assembly language, which performs an iterative or recursive algorithm, such as sorting or searching.
4. Design, code and debug an assembly language program which implements the function calling conventions of a high or mid-level language, such as C.

**Required Textbooks**

The C Programming Language; Brian Kernighan and Dennis Ritchie, Prentice-Hall, 1988.

**Reference Material**

None specified.

## **Major Topics Covered in the Course**

1. Compilation and Module Linkage
2. C syntax and semantics
  - a. Data storage, data types, and variables
  - b. Operators, expressions and statements
  - c. Control structures and control flow design
  - d. Control flow testing and test documentation
  - e. Functions, returned values and parameter lists
  - f. Arrays, character arrays and strings
3. The C standard library
  - a. Standard library character and string functions
  - b. File input/output
4. Assembly Language
  - a. Registers and operations
  - b. Memory addressing and access
  - c. Subprograms and stacks
  - d. Communicating with C programs
5. Comparison between instruction set architectures

## **Laboratory Projects**

Programming Project Area	Weeks
--------------------------	-------

## **CSAB Category Content**

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	0	0	Computer Organization and Architecture	1	0
Algorithm & Software Design	0	0	Concepts of Programming Languages	2	0

## **Oral and Written Communications**

There are no significant oral or written communications required in this course. Virtually all assignments consist of writing computer programs.

### **Social and Ethical Issues**

This course contains no significant coverage of social and ethical issues beyond the usual proscriptions against plagiarism and cheating.

### **Theoretical Content**

There is no significant coverage of theoretical topics in this course.

### **Problem Analysis**

The instructor performs analysis of representative problems in class. All student programming assignments require the student to analyze problem requirements. The instructor analyzes problem solutions (both his own and student's) in class.

### **Solution Design**

This course requires the student to craft 8-10 correctly functioning computer programs. The requirements for each program will necessitate that the student comprehend and apply mathematical knowledge to their program designs. These programs range from 50-100 lines of code at the beginning of the semester, to a final project whose design usually entails >300 lines of program code.

### **CC2001**

This course provides coverage of topics in the following areas (hours listed are minimums):

PF1. Fundamental programming constructs [core]	3
PF2. Algorithms and problem solving [core]	3
PF3. Fundamental data structures [core]	3
AR2. Machine representation of data [core]	1
AR3. Assembly level machine organization [core]	7
AR4. Memory system organization and architecture [core]	5
AR5. Interfacing and communication [core]	3

---

Course Coordinator: Dick Lang  
Last Updated: January 21, 2009 (approval pending)  
Syllabus Version Number: 1.1