

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

CS 121
Program Design and Development
4 Semester Hours
(3 lecture hours, 3 laboratory hours)

Catalog Description

Formulation of problems and top-down design of programs in a modern structured language for their solution on a digital computer.

Prerequisite Courses

Math 107 or Math 201 or a satisfactory math placement score

Prerequisite Topics

- Algebra and linear equations
- Mathematical notation and formalism
- Polynomial, trigonometric, logarithmic and exponential functions

Measured Course Outcomes

Students taking this course will (among other things):

1. Analyze and explain the behavior of simple computer programs involving fundamental programming constructs. (*Contributes to performance criterion A-2*)
2. Design, code, test and debug programs that use the following fundamental programming constructs: basic computation, simple I/O, conditional and iterative control structures, composite data types and function definition and invocation. (*Contributes to performance criterion K-3*)
3. Apply techniques of structured decomposition to break a problem into multiple program components. (*Contributes to performance criterion A-1*)
4. Use multiple strategies or tools to debug a simple program, including code tracing in a debugger. (*Contributes to performance criterion I-4*)

Required Textbooks

H. M. Deitel and T. J. Deitel, C How to Program, Prentice Hall, 2001, ISBN 0-13-058004-X.

or

Mark Guzdial, Introduction to Computing and Programming in Python, Pearson Prentice Hall, 2005, ISBN 0-13-117655-2.

or

Bradley Miller and David Ranum, Problem Solving with Algorithms and Data Structures using Python, Franklin, Beedle & Associates, 2006, ISBN 1-59028-053-9.

or

Jennifer Cambell, Paul Gries, Jason Montojo and Greg Wilson, Practical Programming: An Introduction to Computer Science Using Python, Pragmatic Bookshelf, 2009, ISBN 1-934356-27-1.

Reference Material

Kernighan and Ritchie, C Programming Language, 2nd Edition, Prentice Hall, 1988.

Major Topics Covered in the Course

1. Software development life cycle
2. Input and output functions
3. Data storage, data types, and variables
4. Operators, expressions and statements
5. Control structures and control flow design
6. Control flow testing and test documentation
7. Concepts of modularity, encapsulation, information hiding
8. Problem partitioning and decomposition; top-down design
9. Functions, returned values and parameter lists
10. Arrays, including single and multidimensional arrays, character arrays
11. Standard character and string functions
12. File input/output
13. Linear search, and bubble or exchange sort
14. Code tracing

Laboratory Projects

All programming projects and assignments are to be derived and developed by students individually. In this course, students do not perform assignments as members of teams.

Programming projects will be assigned approximately weekly with the expectation that laboratory time will be used to enable students to individually engage with the instructor or lab assistant to understand and work through programming issues.

CSAB Category Content

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	0	0	Computer Organization and Architecture	0	0
Algorithm & Software Design	1	0	Concepts of Programming Languages	3	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. Some coverage of solution accuracy and stability as they relate to critical software applications and liability are presented.

Theoretical Content

This course contains no significant coverage of theoretical topics.

Problem Analysis

Problem analysis concepts are presented in lecture examples. The instructor performs analysis of various 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 12-14 small, correctly functioning computer programs. The requirements for each program will necessitate that the student comprehend and apply mathematical knowledge to design and implement programs. These programs range from 10-20 lines of code at the beginning of the semester, to a final assignment whose design entails ~200 lines of program code.

CC2001

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

PF1. Fundamental programming constructs [core]	7
PF2. Algorithms and problem solving [core]	3
PF3. Fundamental data structures [core]	3
AR2. Machine representation of data [core]	1

Course Coordinator: Dick Lang
Last Updated: April 15, 2009 (Approved)
Syllabus Version Number: 1.2