

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

CS 224
Programming Tools
2 Semester hours

Catalog Description

Debugging tools, scripting languages, UNIX programming tools, introduction to graphical user interface programming.

Prerequisite Courses

CS 122 – Data Structures

Prerequisite Topics

- Some programming experience.
- Basic data structures.

Measured Course Outcomes

Students taking this course will:

1. Design, code and debug Unix shell scripts (*contributes to performance criterion I-3*).
2. Utilize a source code control utility and the Unix make utility (or ANT) to manage the revisions and configuration of a program consisting of multiple source files (*contributes to performance criterion I-4*).
3. Use a source code debugger to trace execution, examine variables and set breakpoints in the process of debugging a program (*contributes to performance criterion I-4*).
4. Design, code and debug a simple graphical user interface (*contributes to performance criterion I-5*).

Required Textbooks

One of:

- Kernighan and Pike, The UNIX Programming Environment, Prentice Hall, 1984.
- Advanced Programming in the Unix Environment, W. Richard Stevens, Addison-Wesley, 1992, ISBN 0-201-56317-7.

Reference Material

Unix online documentation.

Kernighan and Pike, The Practice of Programming, Addison-Wesley, 1999.

Stones and Matthew, Beginning Linux Programming, 2nd. Ed., Wrox Press, 1999.

Major Topics Covered in the Course

1. UNIX/Linux concepts
2. Basic shell and Unix commands
3. Compilation, linking and make (or ANT)
4. Revision, configuration and source code control
5. Shell scripting
6. Other scripting languages (such as: awk, Perl, Python or Tcl)
7. Using a debugger and other debugging practices
8. Introduction to GUI programming
9. Event-driven programming
10. Interactive GUI design tools

Laboratory Projects

<u>Programming Project Area</u>	<u>Weeks</u>
Shell scripting	2
Programming development using multiple source files	3
Graphical user interface programming	2

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	1	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

This course includes no significant coverage of computer science theory.

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. A portion of the course is devoted to learning skills associated with interactive debugging tools. In effect, students learn to use these tools to analyze their programs.

Solution Design

This course requires the student to craft 6-8 correctly functioning computer programs. The requirements for each program will necessitate that the student comprehend and apply lecture material and online documentation to the design of programs, scripts and GUI interfaces.

CC2001

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

PF5. Event-driven programming [core]	4
HC2. Building a simple graphical user interface [core]	2
SE2. Using APIs [core]	2
SE3. Software tools and environments [core]	3
OS12. Scripting [elective]	3

Course Coordinator: Wayne Cochran
Last Updated: September 16, 2005 (Approved)
Syllabus Version Number: 1.2