Washington State University
School of EECS
Computer Science Course Assessment Report

Course Number CptS 224
Course Title Programming Tools

Semesters Offered Summer Spring
Instructor Andrew O'Fallon
10th Day Enrollment 21 Number Completing Successfully (C grade or better) 20

I. Assessment Outcomes from the Course Syllabus – check those that apply

- ☒ (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline. In particular, students should be able to apply this knowledge in a way that demonstrates comprehension of the tradeoffs involved in the modeling, design and development of software systems of various scales and complexity
- ☐ (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- ☒ (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
- ☐ (d) An ability to function effectively on teams to accomplish a common goal
- ☐ (e) An understanding of professional, ethical, and social responsibilities
- ☐ (f) An ability to communicate effectively
- ☐ (g) An ability to analyze the impact of computing on individuals, organizations, and society, including ethical, legal, security, and global policy issues
- ☐ (h) Recognition of the need for and an ability to engage in continuing professional development
- ☒ (i) An ability to use current techniques, skills, and tools necessary for computing practice.
- ☐

II. List of Course Topics from the Course Syllabus

1. UNIX/Linux Programming: shell commands, make, revision control
2. Scripting Languages: (Bourne) shell scripting, Python (or Perl or awk)
3. Debugging: bug taxonomy, debugging strategies, debugging with printf(), gdb, graphical debugging (xxgdb or ddd)
4. Introduction to Graphical User Interface Programming: event-driven programming, compiled GUIs (FLTK, Qt, Gtk+, or xforms), interactive GUI designers, (FLTK -> FLUID, Qt -> Qt Designer, Gtk+ -> glade, xforms -> fdesign)

III. Course Assessment Summary Table: one row of the table should be devoted to each of the checked outcomes in part I.

Outcome	Topics	Specific Measures (These should be available in the course materials file for inspection.) List specific exams, homeworks, projects, presentations in which students demonstrate their achievement of this outcome

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline. In particular, students should be able to apply this knowledge in a way that demonstrates comprehension of the tradeoffs involved in the modeling, design and development of software systems of various scales and complexity	1-4	Homework 3 – "Make" with Connect Four Application; Homework 4 – Shell Metacharacters and Regular Expressions; Homework 5 – Regular Expressions and Sed; Homework 6 – Awk; Homework 7 – Building a Shell Program that emulates "wc"; Homework 8 – Write a Perl Script that Parses Data from a File; Homework 9 – Apply a GUI Builder to Build an Interface to a Regular Expression Matching Application; Exam 1 – Focuses on solving problems related to the UNIX filesystem (Q 1, 3, 7), basic commands (Q 2, 5, 8), permissions (Q 2), shell patterns and metacharacters (Q 4, 8), make (Q 5), source control (Q 9), debugging (Q 6), and UNIX filters (Q 8) (i.e. grep, sed, sort, etc.) Final Exam – Focuses on solving problems related to basic commands (Q 2, 4, 5), shell patterns and metacharacters (Q 4, 5), make (Q 2), debugging (Q 3), and UNIX filters (Q 4, 5) (i.e. grep, sed, sort, etc.), awk (Q 7), shell scripting (Q 8, 9), and Perl scripting (Q 6, 9)

(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs	1-4	Homework 1 – Basic UNIX Commands; Homework 2 – The UNIX File System; Homework 3 – "Make" with Connect Four Application; Homework 4 – Shell Metacharacters and Regular Expressions; Homework 5 – Regular Expressions and Sed; Homework 6 – Awk; Homework 7 – Building a Shell Program that emulates "wc"; Homework 8 – Write a Perl Script that Parses Data from a File; Homework 9 – Apply a GUI Builder to Build an Interface to a Regular Expression Matching Application; Exam 1 – Focuses on solving problems related to the UNIX filesystem (Q 1, 3, 7), basic commands (Q 2, 5, 8), permissions (Q 2), shell patterns and metacharacters (Q 4, 8), make (Q 5), source control (Q 9), debugging (Q 6), and UNIX filters (Q 8) (i.e. grep, sed, sort, etc.) Final Exam – Focuses on solving problems related to the UNIX filesystem (Q 1), basic commands (Q 2, 4, 5), shell patterns and metacharacters (Q 4, 5), make (Q 2), debugging (Q 3), and UNIX filters (Q 4, 5) (i.e. grep, sed, sort, etc.), awk (Q 7), shell scripting (Q 8, 9), and Perl scripting (Q 6, 9)

(i) An ability to use current techniques, skills, and tools necessary for computing practice.	1-4	Homework 1 – Basic UNIX Commands; Homework 2 – The UNIX File System; Homework 3 – "Make" with Connect Four Application; Homework 4 – Shell Metacharacters and Regular Expressions; Homework 5 – Regular Expressions and Sed; Homework 6 – Awk; Homework 7 – Building a Shell Program that emulates "wc"; Homework 8 – Write a Perl Script that Parses Data from a File; Homework 9 – Apply a GUI Builder to Build an Interface to a Regular Expression Matching Application; Exam 1 – Focuses on solving problems related to the UNIX basic commands (Q 2, 5, 8), permissions (Q 2), shell patterns and metacharacters (Q 4, 8), make (Q 5), source control (Q 9), debugging (Q 6), and UNIX filters (Q 8) (i.e. grep, sed, sort, etc.) Final Exam – Focuses on solving problems related to basic commands (Q 2, 4, 5), shell patterns and metacharacters (Q 4, 5), make (Q 2), debugging (Q 3), and UNIX filters (Q 4, 5) (i.e. grep, sed, sort, etc.), awk (Q 7), shell scripting (Q 8, 9), and Perl scripting (Q 6, 9)

IV. Using the table as a guide, for each outcome summarize your evaluation of the students' achievement of that outcome; cite student performance on the identified measures as evidence to support your conclusions.

You will notice as you read through the following outcomes that the averages of assignments and exams were above passing with a C or better. In many instances the averages were B or higher!

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline. In particular, students should be able to apply this knowledge in a way that demonstrates comprehension of the tradeoffs involved in the modeling, design and development of software systems of various scales and complexity.

Homework 3 – "Make" with Connect Four Application. This assignment allowed students to develop an appropriate dependency graph model of a Connect Four Game Application using Make (Avg. 84.4 %). Homework 4 – Shell Metacharacters and Regular Expressions. This assignment allowed students to use UNIX shell metacharacters and regular expressions to model and extract information about files within a directory (Avg. 94.7 %). Homework 5 – Regular Expressions and Sed. This assignment also required that students represent and model regular expressions (Avg. 72.9 %). Homework 6 – Awk. Students used Awk to perform some modifications to a calendar program. Students had to identify why Awk was the most efficient solution to solving this problem (Avg. 87.5 %). Homework 7 – Building a Shell Program that emulates "wc". Students had to complete a Bourne Shell program that emulated the built-in UNIX wc program. Students had to understand the language benefits that Bourne Shell has as compared to other shells or languages (Avg. 96.8 %). Homework 8 – Write a Perl Script that Parses Data from a File. Students had to model an input file and understand how to parse it using Perl. For this particular application students learned why Perl is more efficient than shell scripts (Avg. 86.3 %). Homework 9 – Apply a GUI Builder to Build an Interface to a Regular Expression Matching Application. This assignment was optional. It required knowledge of the QT GUI builder and QT functions (Avg. 92.5 %).

Exam 1 – Focuses on solving problems related to the UNIX filesystem (Q 1, 3, 7), basic commands (Q 2, 5, 8), permissions (Q 2), shell patterns and metacharacters (Q 4, 8), make (Q 5), source control (Q 9), debugging (Q 6), and UNIX filters (Q 8) (i.e. grep, sed, sort, etc.). Avg. 80.1 %. Question really required that students recognized which UNIX commands and filters were best for solving some specified problem.

Final Exam – Focuses on solving problems related to basic commands (Q 2, 4, 5), shell patterns and metacharacters (Q 4, 5), make (Q 2), debugging (Q 3), and UNIX filters (Q 4, 5) (i.e. grep, sed, sort, etc.), awk (Q 7), shell scripting (Q 8, 9), and Perl scripting (Q 6, 9). Avg. 76.9 %.

(b) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

Note: For some of the averages, please refer to (a) above.

There were multiple assignments related to topics 1-4. Each new assignment usually required knowledge of previous topics. Thus, there was a continual refresh of topics as students completed new assignments. I tried to make it such that new assignments approached the topic from a different view than previous assignments.

Both homework and exams tested students' abilities to comprehend and apply the topics presented in this course. The homework provided usually focused on a single topic that we discussed in the course. The assignments required that students applied some basic UNIX commands or script files. The exams brought together all of the topics. The average student successfully completed all assignments and exams.**Error! Not a valid bookmark self-reference.**

(c) An ability to use current techniques, skills, and tools necessary for computing practice.

See all average either in (a) or (b) above.

CptS 224 is a UNIX tool class. All homework and exams required the use of the UNIX environment and its commands or tools.

Homework 1 – Basic UNIX Commands; Homework 2 – The UNIX File System; Homework 3 – "Make" with Connect Four Application; Homework 4 – Shell Metacharacters and Regular Expressions; Homework 5 – Regular Expressions and Sed; Homework 6 – Awk; Homework 7 – Building a Shell Program that emulates "wc"; Homework 8 – Write a Perl Script that Parses Data from a File; Homework 9 – Apply a GUI Builder to Build an Interface to a Regular Expression Matching Application;

Exam 1 – Focuses on solving problems related to the UNIX basic commands (Q 2, 5, 8), permissions (Q 2), shell patterns and metacharacters (Q 4, 8), make (Q 5), source control (Q 9), debugging (Q 6), and UNIX filters (Q 8) (i.e. grep, sed, sort, etc.)

Final Exam – Focuses on solving problems related to basic commands (Q 2, 4, 5), shell patterns and metacharacters (Q 4, 5), make (Q 2), debugging (Q 3), and UNIX filters (Q 4, 5) (i.e. grep, sed, sort, etc.), awk (Q 7), shell scripting (Q 8, 9), and Perl scripting (Q 6, 9)There were multiple assignments related to topics 1-4. Each new assignment usually required knowledge of previous topics. Thus, there was a continual refresh of topics as students completed new assignments. I tried to make it such that new assignments approached the topic from a different view than previous assignments.

Both homework and exams tested students' abilities to comprehend and apply the topics presented in this course. The homework provided usually focused on a single topic that we discussed in the course. The assignments required that students applied some basic UNIX commands or script files. The exams brought together all of the topics. The average student successfully completed all assignments and exams.**Error! Not a valid bookmark self-reference.**

V. Qualitative Assessment of Student Performance: using the arguments above and other data support the claim that students who completed this course with a grade of C or better have achieved each of the intended outcomes of this course.

There were multiple assignments related to topics 1-4. Each new assignment usually required knowledge of previous topics. Thus, there was a continual refresh of topics as students completed new assignments. I tried to make it such that new assignments approached the topic from a different view than previous assignments.

Both homework and exams tested students' abilities to comprehend and apply the topics presented in this course. The homework provided usually focused on a single topic that we discussed in the course. The assignments required that students applied some basic UNIX commands or script files. The exams brought together all of the topics. The average student successfully completed all assignments and exams.**Error! Not a valid bookmark self-reference.**

VI. Concerns: state any concerns you may hold about this class – were the students adequately prepared coming into it? Are there topics or outcomes where (some) students were weak after completing the course? Other concerns? Were there any comments on students' course evaluations that should be addressed in future instances of the course?

This section is very important for improving our program: it provides critical input to the curriculum committee for identifying areas requiring attention.

This class is a 2 credit course and yet we try to cover as much material as would be required for a 3 credit course. Focusing on fewer topics, but getting more in depth with these topics would probably be ideal. Placing more effort and practice towards the scripting languages would ultimately be beneficial to the students.

[Note: A "meta-concern" already noted by the course coordinator in preparing the above table is that the mapping of topics to specific measures is somewhat nebulous, as pretty much every topic maps to all ABET outcomes, coverage of a topic usually incorporating knowledge of computing (a), implementation (c), and state-of-the-art techniques to accomplish these (i). Nevertheless, the mapping of specific measures to outcomes is still useful.]

Signature ___Andrew O'Fallon_____ Date: _____5/8/07_____

Please email a copy of the completed form to Patricia Arnold, patricia@eecs.wsu.edu and deliver a signed hardcopy to her mailbox.