UNIVERSITY OF KENTUCKY

New Course Report

4/27/2015 12:46:13 PM

1. General Information

1a. Submitted by the College of: ENGINEERING

Date Submitted: 12/9/2014

1b. Department/Division: Computer Science

1c. Contact Person

Name: Jerzy Jaromczyk

Email: jurek@cs.uky.edu

Phone: 257-1186

Responsible Faculty ID (if different from Contact)

Name: James Griffioen

Email: griff@netlab.uky.edu

Phone: 257-6746

1d. Requested Effective Date: Semester following approval

1e. Should this course be a UK Core Course? No

2. Designation and Description of Proposed Course

2a. Will this course also be offered through Distance Learning?: No

2b. Prefix and Number: CS 270

2c. Full Title: Systems Programming

2d. Transcript Title:

2e. Cross-listing:

2f. Meeting Patterns

LECTURE: 3

2g. Grading System: Letter (A, B, C, etc.)

2h. Number of credit hours: 3

2i. Is this course repeatable for additional credit? No

If Yes: Maximum number of credit hours:

If Yes: Will this course allow multiple registrations during the same semester?

2j. Course Description for Bulletin: This course provides an introduction to computer systems and explores computer architecture, operating systems, and networks from a programmer's perspective. The course also introduces advanced programming and debugging tools. Topics include hardware instruction sets, machine language and C language program representations, linking/loading, operating systems (process management, scheduling, memory management, interprocess communication, and file systems), network programming (socket programming and web protocols), and common security attacks and solutions.

2k. Prerequisites, if any: EE 280 and CS 216

2l. Supplementary Teaching Component:

3. Will this course taught off campus? No

If YES, enter the off campus address:

4. Frequency of Course Offering: Spring,

Will the course be offered every year?: Yes

If No, explain:

5. Are facilities and personnel necessary for the proposed new course available?: Yes

If No, explain:

6. What enrollment (per section per semester) may reasonably be expected?: 40

7. Anticipated Student Demand

Will this course serve students primarily within the degree program?: Yes

Will it be of interest to a significant number of students outside the degree pgm?: Yes

If Yes, explain: It will also be of interest to Computing Engineering majors and some Electrical Engineering majors.

8. Check the category most applicable to this course: Relatively New – Now Being Widely Established,

If No, explain:

9. Course Relationship to Program(s).

a. Is this course part of a proposed new program?: No

If YES, name the proposed new program:

b. Will this course be a new requirement for ANY program?: No

If YES, list affected programs:

10. Information to be Placed on Syllabus.

a. Is the course 400G or 500?: No

b. The syllabus, including course description, student learning outcomes, and grading policies (and 400G-/500-level grading differentiation if applicable, from **10.a** above) are attached: Yes

Distance Learning Form

Instructor Name:

Instructor Email:

Internet/Web-based: No

Interactive Video: No

Hybrid: No

1.How does this course provide for timely and appropriate interaction between students and faculty and among students? Does the course syllabus conform to University Senate Syllabus Guidelines, specifically the Distance Learning Considerations?

2.How do you ensure that the experience for a DL student is comparable to that of a classroom-based student's experience? Aspects to explore: textbooks, course goals, assessment of student learning outcomes, etc.

3.How is the integrity of student work ensured? Please speak to aspects such as password-protected course portals, proctors for exams at interactive video sites; academic offense policy; etc.

4.Will offering this course via DL result in at least 25% or at least 50% (based on total credit hours required for completion) of a degree program being offered via any form of DL, as defined above?

If yes, which percentage, and which program(s)?

5.How are students taking the course via DL assured of equivalent access to student services, similar to that of a student taking the class in a traditional classroom setting?

6.How do course requirements ensure that students make appropriate use of learning resources?

7.Please explain specifically how access is provided to laboratories, facilities, and equipment appropriate to the course or program.

8.How are students informed of procedures for resolving technical complaints? Does the syllabus list the entities available to offer technical help with the delivery and/or receipt of the course, such as the Information Technology Customer Service Center (http://www.uky.edu/UKIT/)?

9.Will the course be delivered via services available through the Distance Learning Program (DLP) and the Academic Technology Group (ATL)? NO

If no, explain how student enrolled in DL courses are able to use the technology employed, as well as how students will be provided with assistance in using said technology.

10.Does the syllabus contain all the required components? NO

11.I, the instructor of record, have read and understood all of the university-level statements regarding DL.

Instructor Name:

SIGNATURE|SEALES|William B Seales|CS 270 NEW Dept Review|20131118

SIGNATURE|BJSTOK0|Barbara J Brandenburg|CS 270 NEW College Review|20140207

SIGNATURE|SEALES|William B Seales|CS 270 ZCOURSE_NEW Approval Returned to Dept|20140210

SIGNATURE|BJSTOK0|Barbara J Brandenburg|CS 270 NEW College Review|20140220

SIGNATURE|SEALES|William B Seales|CS 270 ZCOURSE_NEW Approval Returned to Dept|20140220

SIGNATURE|BJSTOK0|Barbara J Brandenburg|CS 270 NEW College Review|20140221

SIGNATURE|JMETT2|Joanie Ett-Mims|CS 270 NEW Undergrad Council Review|20150427

SIGNATURE|SEALES|William B Seales|CS 270 NEW Dept Review|20141210

SIGNATURE|BJSTOK0|Barbara J Brandenburg|CS 270 NEW College Review|20150227

SIGNATURE|JMETT2|Joanie Ett-Mims|CS 270 NEW Undergrad Council Review|20150427

New Course Form

https://myuk.uky.edu/sap/bc/soap/rfc?services=

Open in full window to print or save

Generate R

Attachments:

Browse... Upload File

	ID	Attachment
Delete	2454	CS270-Faculty-Support-2013.pdf
Delete	4899	cs270-syllabus.pdf

First 1 Last

(*denotes required fields)

1. **General Information**

a. * Submitted by the College of: ENGINEERING Submission Date: 12/9/2014

b. * Department/Division: Computer Science

c.

* Contact Person Name:	Jerzy Jaromczyk	Email: jurek@cs.uky.edu	Phone: 257-1186
* Responsible Faculty ID (if different from Contact)	James Griffioen	Email: griff@netlab.uky.edu	Phone: 257-6746

d. * Requested Effective Date: ● Semester following approval OR ○ Specific Term/Year [1]

e. Should this course be a UK Core Course? ○ Yes ● No

If YES, check the areas that apply:

□ Inquiry - Arts & Creativity

□ Inquiry - Humanities

□ Inquiry - Nat/Math/Phys Sci

□ Inquiry - Social Sciences

□ Composition & Communications - I

□ Composition & Communications - II

□ Quantitative Foundations

□ Statistical Inferential Reasoning

□ U.S. Citizenship, Community, Diversity

□ Global Dynamics

2. **Designation and Description of Proposed Course.**

a. * Will this course also be offered through Distance Learning? ○ Yes [4] ● No

b. * Prefix and Number: CS 270

c. * Full Title: Systems Programming

d. Transcript Title (if full title is more than 40 characters):

e. To be Cross-Listed [2] with (Prefix and Number):

f. * Courses must be described by at least one of the meeting patterns below. Include number of actual contact hours[3] for each meeting pattern type.

3	Lecture		Laboratory[1]		Recitation		Discussion
	Indep. Study		Clinical		Colloquium		Practicum
	Research		Residency		Seminar		Studio
	Other		If Other, Please explain:				

g. * Identify a grading system:

● Letter (A, B, C, etc.)
○ Pass/Fail
○ Medicine Numeric Grade (Non-medical students will receive a letter grade)
○ Graduate School Grade Scale

h. * Number of credits: 3

i. * Is this course repeatable for additional credit? ○ Yes ● No
If YES: Maximum number of credit hours:
If YES: Will this course allow multiple registrations during the same semester? ○ Yes ○ No

j. * Course Description for Bulletin:

This course provides an introduction to computer systems and explores computer architecture, operating systems, and networks from a programmer's perspective. The course also introduces advanced programming and debugging tools. Topics include hardware instruction sets, machine language and C language program representations, linking/loading, operating systems (process management, scheduling, memory management, interprocess communication, and file systems), network programming (socket programming and web protocols), and common security attacks and solutions.

k. Prerequisites, if any:

EE 280 and CS 216

l. Supplementary teaching component, if any: ○ Community-Based Experience ○ Service Learning ○ Both

3. *** Will this course be taught off campus?** ○ Yes ● No

If YES, enter the off campus address:

4. **Frequency of Course Offering.**

a. * Course will be offered (check all that apply): ☑ Fall ☑ Spring □ Summer □ Winter

b. * Will the course be offered every year? ● Yes ○ No

If No, explain:

5. *** Are facilities and personnel necessary for the proposed new course available?** ● Yes ○ No

If No, explain:

6. *** What enrollment (per section per semester) may reasonably be expected?** 40

7. **Anticipated Student Demand.**

a. * Will this course serve students primarily within the degree program? ● Yes ○ No

b. * Will it be of interest to a significant number of students outside the degree pgm? ● Yes ○ No

If YES, explain:

It will also be of interest to Computing Engineering majors and some Electrical Engineering majors.

8. *** Check the category most applicable to this course:**

□ Traditional – Offered in Corresponding Departments at Universities Elsewhere
☑ Relatively New – Now Being Widely Established
□ Not Yet Found in Many (or Any) Other Universities

9. **Course Relationship to Program(s).**

a. * Is this course part of a proposed new program? ○ Yes ● No

If YES, name the proposed new program:

b. * Will this course be a new requirement [5]for ANY program? ○ Yes ● No

If YES[5], list affected programs::

10. **Information to be Placed on Syllabus.**

a. * Is the course 400G or 500? ○ Yes ● No

If YES, the *differentiation for undergraduate and graduate students must be included* in the information required in **10.b**. You must include: (i) identi additional assignments by the graduate students; and/or (ii) establishment of different grading criteria in the course for graduate students. (See SR

b. ☑ * The syllabus, including course description, student learning outcomes, and grading policies (and 400G-/500-level grading differentiation if appl **10.a** above) are attached.

[1] Courses are typically made effective for the semester following approval. No course will be made effective until all approvals are received.

[2] The chair of the cross-listing department must sign off on the Signature Routing Log

[3] In general, undergraduate courses are developed on the principle that one semester hour of credit represents one hour of classroom meeting per week for a semester, exclusive of any laboratory meeting. Laboratory meeting, generally, re two hours per week for a semester for one credit hour. (from SR 5.2.1)

[4] You must also submit the Distance Learning Form in order for the proposed course to be considered for DL delivery.

[5] In order to change a program, a program change form must also be submitted.

Rev 8/09

UK
UNIVERSITY OF
KENTUCKY

College of Engineering
Department of Computer Science
329 Rose Street
Davis Marksbury Building
Lexington, KY 40506-0633

859 257-3961

www.cs.uky.edu

October 15, 2013

MEMORANDUM

TO: Jim Griffioen
FROM: Brent Seales W/BS
SUBJECT: New Course Proposal CS 270

The new course proposal for CS 270: Systems Programming was vetted by the Computer Science faculty on Monday, 14 October 2013. After discussion by the faculty, the proposal was approved by unanimous vote for submission to the next level in the approval process.

CS 270 Systems Programming

University of Kentucky
Department of Computer Science
Semester and Year

Course Information

Meeting Times

Days and Time: Room Number

Instructor

Instructor Name, Office Number, phone, email address

Office Hours

Days and Times: Room Number
You can also send questions via email.

Course Description

This course provides an introduction to computer systems and explores computer architecture, operating systems, and networks from a programmer's perspective. The course also introduces advanced programming and debugging tools. Topics include hardware instruction sets, machine language and C language program representations, linking/loading, operating systems (process management, scheduling, memory management, interprocess communication, and file systems), network programming (socket programming and web protocols), and common security attacks and solutions.

Course Objectives

The goal of the course is to provide an overall view of the components that make up a computer system and the ways in which programs interact with those components. The course touches on a broad range of topics including hardware, assembly language, compilers, operating systems, networking, and graphical users interfaces. Topics are presented from a programmer's perspective with the goal of developing a complete understanding of what happens when a program is excecuted. The course also introduces useful programming and debugging tools that can assist in creating robust and efficient code. The course provides foundational progamming skills required by many upper level classes.

Hands-on projects and in-class assignments will make the concepts real and expose students to the challenges of developing robust and efficient software.

Learning Outcomes

Students will acquire knowlege of computer systems and how to program them. More specifically, students will be able to

1. Demonstrate and understanding of overall computer system structure and operating system design.
2. Demonstrate an understanding of how specific high-level language program constructs are implemented in assembly language.
3. Be able to effectively use a debugger.
4. Demonstrate an understanding the concepts of processes and threads and use them to achieve concurrent execution in applications.
5. Demonstrate and understanding of the interfaces provided by the memory management component of the operating system.
6. Use the interfaces provided by the file system in developing applications.
7. Use a networking API to create networked applications

Measures of Learning Outcomes

The above outcomes will be evaluated based on the programming projects, in-class/computer lab assignments, and exams. They will also be evaluated based on the students' self-assessment of their mastery of these outcomes performed at the end of the semester.

Course Prerequisites

The course assumes a basic understanding of computer hardware. It also assumes a basic level of proficiency in the C or C++ programming language. Students must have completed EE 280. Students should also have taken CS 216.

Textbook (Required)

The following textbook is required for the course:

- Computer Systems: A Programmer's Perspective (2nd edition), Randal E. Bryant and David R. O'Hallaron, Pearson, 2011, ISBN 10: 0-13-610804-0, ISBN 13: 978-0-13-610804-7
  (The webpage for the book can be found at http://csapp.cs.cmu.edu/).

Exams

- **Exams:** There will be three exams:
  - Exam 1: Day, Date, Time (In-class)
  - Exam 2: Day, Date, Time (In-class)
  - Final Exam: Day, Date, Time (Scheduled Exam Slot)

- **Programming Projects:**

  Programming projects will be given regularly throughout the semester (roughly every three weeks, although the exact dates and timeframes will vary depending on the particular project). The goal of the projects is to reinforce the concepts and material discussed in class. All projects must be implemented using the computing system and development environment specified in the project description. In some cases, the development environment may be constrained for the purposes of teaching a particular programming skill.

  To provide hands-on experience working with various software tools, the Friday class period will often be used for in-class assignments and will take place in the computer lab (Room Number NNN).

  In general, most projects and in-class assignments will be based on the C programming language and the Linux development environment. Students are expected to already know how to program in C/C++. Please consult http://www.cs.uky.edu/%7eraphael/programming.html for a list of programming skills that students should have familiarity with prior to taking the course. Unless specified otherwise, projects will be done individually by each student. Students must also read and adhere to good programming standards (see http://www.cs.uky.edu/%7eraphael/checklist.html for a checklist of good programming characteristics). Programs will be penalized for poor programming style.

  All programming projects and in-class assignment will be submitted electronically via the CS Portal: https://www.cs.uky.edu/csportal. An explanation of how to submit your work will be given with each project.

  The class textbook has an associated "student site" that can be found at http://csapp.cs.cmu.edu/public/students.html. The student site has pointers to helpful resources that better describe the C languange and Linux operating system/programming environment. Additional resources for specific chapters in the textbook are also available on the student site.

Policies

Grading

Grades will be determined based on programming assignments (50%), exams (45% = 15% exam 1 + 15% exam 2 + 15% final exam), and in-class/computer lab assignments (5%). Final grades will be assigned according to the following scale:

A=90-100%, B=80-89%, C=70-79%, D=60-69%, E=0-59%.

No incomplete grades will be assigned unless there exist exceptional, extenuating circumstances.

Late Penalty

Late programs will be penalized 10% (of the total possible points) per day for each day late not including weekend or holiday days. No project may be submitted more than five days late not including weekend or holiday days.

Academic Integrity

Per university policy, students shall not plagiarize, cheat, or falsify or misuse academic records. Students are expected to adhere to University policy on cheating and plagiarism in all courses. The minimum penalty for a first offense is a zero on the assignment on which the offense occurred. If the offense is considered severe or the student has other academic offenses on their record, more serious penalties, up to suspension from the university may be imposed.

Plagiarism and cheating are serious breaches of academic conduct. Each student is advised to become familiar with the various forms of academic dishonesty as explained in the Code of Student Rights and Responsibilities. Complete information can be found at the following website: http://www.uky.edu/Ombud. A plea of ignorance is not acceptable as a defense against the charge of academic dishonesty. It is important that you review this information as all ideas borrowed from others need to be properly credited.

Part II of *Student Rights and Responsibilities* (available online http://www.uky.edu/StudentAffairs/Code/part2.html) states that all academic work, written or otherwise, submitted by students to their instructors or other academic supervisors, is expected to be the result of their own thought, research, or self-expression. In cases where students feel unsure about the question of plagiarism involving their own work, they are obliged to consult their instructors on the matter before submission.

When students submit work purporting to be their own, but which in any way borrows ideas, organization, wording or anything else from another source without appropriate acknowledgement of the fact, the students are guilty of plagiarism. Plagiarism includes reproducing someone elses work, whether it be a published article, chapter of a book, a paper from a friend or some file, or something similar to this. Plagiarism also includes the practice of employing or allowing another person to alter or revise the work which a student submits as his/her own, whoever that other person may be.

Students may discuss assignments among themselves or with an instructor or tutor, but when the actual work is done, it must be done by the student, and the student alone. When a students assignment involves research in outside sources of information, the student must carefully acknowledge exactly what, where and how he/she employed them. If the words of someone else are used, the student must put quotation marks around the passage in question and add an appropriate indication of its origin. Making simple changes while leaving the organization, content and phraseology intact is plagiaristic. However, nothing in these Rules shall apply to those ideas which are so generally and freely circulated as to be a part of the public domain (Section 6.3.1).

Please note: Any assignment you turn in may be submitted to an electronic database to check for plagiarism.

Attendance Policy

Daily attendance is not taken; however, you are nevertheless expected to attend daily lectures. The number one factor that correlates with success in engineering is going to class. Attendance in lecture *is required* on the days of in-class exams. In case of a legitimate unforeseen conflict (for example, illness or death in the immediate family), if the student contacts the instructor within a few days of the test date and provides proof of the legitimate excuse, the student may take the exam (or a substitute exam) at a later time.

In case of other conflicts, the student must inform the instructor in advance as far as possible, and must take the exam before the scheduled time. Advance notification of such a conflict does not automatically guarantee a special exam will be scheduled; the instructor will make the final decision.

Excused Absences

Students need to notify the professor of absences prior to class when possible. S.R. 5.2.4.2 defines the following as acceptable reasons for excused absences: (a) serious illness, (b) illness or death of family member, (c) University-related trips, (d) major religious holidays, and (e) other circumstances found to fit "reasonable cause for nonattendance" by the professor.

Students anticipating an absence for a major religious holiday are responsible for notifying the instructor in writing of anticipated absences due to their observance of such holidays no later than the last day in the semester to add a class.

Verification of Absences

Students may be asked to verify their absences in order for them to be considered excused. Senate Rule 5.2.4.2 states that faculty have the right to request "appropriate verification" when students claim an excused absence because of illness or death in the family. Appropriate notification of absences due to university-related trips is required prior to the absence.

Making up Missed Work

Students missing any graded work due to an excused absence bear the responsibility of informing the Instructor about their excused absence within one week following the period of the excused absence (except where prior notification is required), and of making up the missed work. Work missed due to an excused absence should be made up, if feasible, before the end of the semester. Students should meet with the instructor to devise a plan and establish a timeline for completing makeup work.

Accommodations due to disability

If you have a documented disability that requires academic accommodations, please see me as soon as possible during scheduled office hours. In order to receive accommodations in this course, you must provide me with a Letter of Accommodation from the Disability Resource Center (Room 2, Alumni Gym, 2572754, email address jkarnes@email.uky.edu) for coordination of campus disability services available to students with disabilities.

Additional References

The following books are not required, but you may find them helpful if you feel you could use additional sources of information:

- Structured Computer Organization,5/e, Andrew S. Tannenbaum, Prentice Hall, 2006
- The C Programming Language, Kernighan and Richie, Prentice Hall, Second Edition, 1988.
- The C++ Programming Language, B. Stroustrup, Addison Wesley, Third Edition, 1997.
- The Unix Programming Environment, Brian Kernighan and Rob Pike, Prentice-Hall, 1984.
- The Linux Kernel Primer, Rodriguez, Fischer, and Smolski, Prentice Hall, 2006
- Operating Systems: Internals and Design Principles, 6e, William Stallings, Prentice Hall, 2009.
- Modern Operating Systems,3e, Andrew S. Tanenbaum, Prentice Hall, 2008.
- Operating Systems Design and Implementation, 3/e, Andrew S. Tanenbaum, Prentice Hall, 2006.

- Operating Systems Vade Mecum, Raphael Finkel, Prentice Hall, Second Edition, 1988.
- Operating System Design: The Xinu Approach, Douglas Comer, Prentice Hall, 1984

Schedule

The following is a tentative outline of the material that will be covered each week. Specific reading assignments to accompany the lectures will be assigned throughout the semester.

(1) Course Overview, Introduction to Machine Data Representation
(2) Machine Level Representation of Programs: Basic Programs and Control
(3) Using Programming Tools: Compilers, Debuggers, Disassemblers
(4) Machine Level Representation of Programs: Procedures and Data Structures
(5) Linking and Loading
(6) Memory Management: Physical Memory, Allocation/Reclamation, Swapping
(7) Processes: System Calls, Process Control and Concurrent Execution
(8) Process Management: Processes, Threads, and Scheduling
(9) Virtual Memory: Address Translation, Page Replacement
(10) File Systems: APIs, Naming, Access Control, and File Structures
(11) Graphic User Interfaces: Windowing Systems and Devices
(12) Internetworking: Internet Protocol
(13) Network Programming: Network API Calls
(14) Web Services: Client/Server and Web Protocols
(15) Concurrent Programming: Parallel Constructs, Synchronization
(16) Miscellaneous: Virtual Machines, the Cloud, Security