

Rowan University
Computer Science Department

Proposal For A New Non-General Education Course: Concurrent Programming

1. Details

a. Course Title

0704.391 Concurrent Programming.

b. Sponsor

Stephen J. Hartley and the Computer Science Department. The primary sponsor will be responsible for administration of the course.

c. Semester Hours

3 semester hours.

d. Course Level

Junior (300 level). General Education designation is not sought.

e. Curricular Effect

The proposed course will be added to the “Restricted Electives” bank of the computer science major and minor, increasing the number and variety of those electives. We hope this will entice more students to major or minor in computer science.

No course will be deleted from the curriculum as a result of adding this course. No changes in faculty loads are anticipated. The enrollments in other advanced electives might decrease as a result of adding this course; however, this is outweighed by the positives of increasing the number and variety of electives available.

This course will be added to the Networking and Operating Systems specialization.

f. Prerequisites

0704.390 Operating Systems.

g. Time and Scale of Implementation

Available every other year starting in the 2004–2005 academic year.

h. Adequacy of Resources

The current staff of the Computer Science department and available computing resources are adequate to implement the proposed course.

i. Recommended Library Resources

Library resources need to include the following. Missing items (see attached Library Resources form) need to be purchased.

Gregory R. Andrews, *Foundations of Multithreaded, Parallel, and Distributed Programming*, Addison Wesley, 2000.

Ken Arnold, James Gosling, and David Holmes, *The Java Programming Language*, 3rd ed, Addison Wesley, 2000.

Stephen J. Hartley, *Concurrent Programming: The Java Programming Language*, Oxford University Press, 1998.

Allen Holub, *Taming Java Threads*, APress, 2000.

Cay S. Horstmann and Gary Cornell, *Core Java*, Volume I and II, Prentice Hall, 2003.

Paul Hyde, *Java Thread Programming*, Sams, 1999.

Doug Lea, *Concurrent Programming in Java*, 2nd ed., Addison Wesley, 2000.

Bil Lewis and Dan Berg, *Multithreaded Programming with Java*, SunSoft Press, 1999.

Jeff Magee and Jeff Kramer, *Concurrency: State Models and Java Programs*. Wiley, 1999.

Scott Oaks and Henry Wong, *Java Threads*, 2nd ed., O'Reilly, 1999.

j. Short-Term Evaluation

The Department Curriculum Committee routinely evaluates all courses to confirm that they meet the standards of the Department, the College of Liberal Arts and Sciences, and the University.

2. Rationale

The typical undergraduate operating systems course includes a component on processes, threads, and synchronization tools because the operating system must implement and support them. Practice using the synchronization tools in a programming language that supports multiple threads might also be included, time permitting. This technique is called *concurrent* or *multithreaded* programming.

I have twice tried to incorporate this material, including an introduction to multithreaded programming using Java, into the Operating Systems course at Rowan and have decided that it is too much to squeeze in and too challenging at that level. I believe this material belongs in an advanced elective course at Rowan for which Operating Systems is a prerequisite.

3. Essence of the Course

a. Objectives in Relation to Student Outcomes

Upon completion of Concurrent Programming, successful students will understand

- the history and benefits/advantages of multithreaded programming;
- several example applications of multithreaded programming;
- the two parallel computer architectures called shared-memory multiprocessors and distributed-memory multicomputers;
- processes, threads, context switching, atomic instructions/actions, shared data, race conditions, critical sections, mutual exclusion, synchronization, locks, barriers, semaphores, and monitors in the context of concurrent programming on a uniprocessor or shared-memory multiprocessor;
- interprocess communication, message passing, remote procedure call, and the rendezvous in the context of distributed programming on a distributed-memory multicomputer.

Successful students will develop concurrent programming skills by using a language that supports the multithreaded paradigm.

b. Topical Outline/Content

History, benefits, and advantages of concurrent programming.

Example multithreaded applications.

Review of uniprocessor computer hardware basics.

Review of processes, time slicing, context switching, process table.

Review of sequential Java.

Parallel computers: shared-memory multiprocessor computer architecture.

Parallel computers: distributed-memory multicomputer architecture.

Threads and their implementation in an operating system.

Java threads.

Shared data and race conditions.

Atomic actions and atomic instructions.

Critical sections and mutual exclusion.

Software solutions: flags and busy waiting.

Hardware solutions: test-and-set.

Theory of locks, semaphores, and barriers.

Java semaphores and examples.

Solving classical synchronization problems with semaphores.

Theory of monitors.

Signaling disciplines.

Java monitors.

Implementation of semaphore with monitor.

Classical solutions using monitors.

Interprocess communication: message passing, remote procedure call, rendezvous.

Implementation in Java.

Classical solutions.

c. Evaluation and Grading

Students will be evaluated by traditional methods such as homework, quizzes, written exams, and programming assignments.

d. Course Evaluation

The courses will be evaluated through customary student evaluations as well as regular departmental review.

4. Results of Consultations

We solicited letters of consultation from Mathematics, Management Information Systems, and Electrical/Computer Engineering. Received letters are attached.

From hartley Tue Jul 29 14:56:25 2003
To: czochor@rowan.edu, hamilton@rowan.edu, head@rowan.edu
Subject: Letters of consultation for two proposals

The Computer Science department will be submitting the following two curriculum proposals this October and we solicit letters of consultation from you. Both proposals are available on the Web as PDF files. Please let me know if you have trouble accessing the documents or have any questions. Thanks for your help.

Minor curricular change to the specializations requirements:

<http://users.rowan.edu/~hartley/changeSpecializations.pdf>

New course proposal, Concurrent Programming:

<http://users.rowan.edu/~hartley/concProgNewCourse.pdf>

From hamilton@rowan.edu Tue Jul 29 15:08:23 2003
Date: Tue, 29 Jul 2003 15:08:06 -0400
From: "Diane Hamilton" <hamilton@rowan.edu>
To: <hartley@elvis.rowan.edu>
Subject: Re: Letters of consultation for two proposals

Steve,

Let me know if you need a letter other than this email. I have looked at the curriculum changes and these changes will not affect the MIS program in any way. Therefore, we have no problems with these proposed changes.

Diane Hamilton, Ph.D.
Professor of MIS
856-256-4760 (phone)
856-256-4439 (fax)

From Czochor@rowan.edu Tue Jul 29 16:23:26 2003
Date: Tue, 29 Jul 2003 16:23:09 -0400
From: "Ronald Czochor" <Czochor@rowan.edu>

revised January 12, 2004

To: <hartley@elvis.rowan.edu>
Subject: Re: Letters of consultation for two proposals

Stephen,

Thank you for the opportunity to comment on your curriculum proposals.

With respect to the Concurrent Programming course, I do not think the objectives give enough information to determine whether there is enough material here for a three credit course. I would like to see a topical outline that describes how parallel processors both in pairs and in multicomputer settings would be studied using threads, context switching, etc. My main questions are how is the course structured and when do particular topics come in?

Ron Czochoch, Chairman
Mathematics Department
Rowan University
Phone: 856.256.4845
Fax 856.256.4816

From hartley Tue Jul 29 16:54:41 2003
To: Czochoch@rowan.edu
Subject: Re: Letters of consultation for two proposals

Thanks, Ron, for your comments. I'll revise the concurrent programming proposal and let you know when it is ready.

From Czochoch@rowan.edu Mon Aug 4 09:56:31 2003
Date: Mon, 04 Aug 2003 09:56:23 -0400
From: "Ronald Czochoch" <Czochoch@rowan.edu>
To: <hartley@elvis.rowan.edu>
Subject: Re: new course proposal

Steve,

I have reviewed the new edition of your proposal for Concurrent Programming and can now give you my unconditional approval. This looks like a substantial course that addresses a demonstrated need in your overall curriculum. Will it be available for students minoring in Computer Science? If so, I'm sure it would be of interest to some of our majors.

Ron Czochoch, Chairman
Mathematics Department

From hartley Mon Aug 4 18:28:49 2003
To: Czochoch@rowan.edu
Subject: Re: new course proposal

Yes, definitely available to anybody --- major, minor, others --- who satisfies the prerequisite. Thanks.

Rowan University

revised January 12, 2004

MEMORANDUM

DATE: 8/4/03

TO : Stephen J. Hartley, Chair
Curriculum Committee
Department of Computer Science

FROM: Linda M. Head, Acting Chair
Electrical and Computer Engineering

RE: New course proposal, Concurrent Programming

The Electrical and Computer Engineering program supports the addition of the new course, Concurrent Programming, to the Computer Science curriculum. We are pleased to have additional opportunities for our undergraduate students who are also CS majors (or minors who meet the pre-requisites) to expand their expertise in this area.

0704.391 **3 s. h.**

Concurrent Programming

(Prerequisites: 0704.390 Operating Systems)

Introduces the motivation for and fundamental concepts of concurrent programming. Topics include processes, threads, context switching, atomic instructions/actions, shared data, race conditions, critical sections, mutual exclusion, synchronization, locks, barriers, semaphores, monitors, shared-memory multiprocessors, and an overview of distributed programming (distributed-memory multicomputers, interprocess communication, message passing, remote procedure call, rendezvous). The course includes developing concurrent programming skills by using a language that supports the multithreaded paradigm.