Glassboro State College Senate Curriculum Committee

# Approval Form

Proposal Title: Foundations of Computer Science

Sponsor(s) Nancy Tinkham Dept.: Computer Science ☎Ext. 7323

**Check one:** ☒ Course ☐ Specialization ☐ Concentration ☐ Minor ☐ Achievement Certificate

☐ Certification Program ☐ Major Program ☐ Minor Change _____ (please name deletion or credit/title/catalog change)

☒ Undergraduate ☐ Graduate ____ Credit Hours

**Step 1 (Department)**

☒ Approved 1/27/93 Date

☐ Not Approved

A. Michael Berman
Dept. CC Chairperson

☒ Reviewed _____ Date

_____
Dept. Chairperson

**Step 2 (Receipt)**

☐ SCC# _____

Proposal Received _____ Date

Mary L. Tuttman
SCC Chairperson

**Step 3 (School CC)**

Reviewed 4-28-93

☒ Approved

☐ Not Approved

**Comments:**

Caldwell
School Curr. Comm. Chairperson

**Step 4 (Academic Dean)**

☐ Recommend

☐ Not Recommend

☐ Conditionally Recommend (see comments)

Reviewed _____ Date

**Comments:**

_____
Signature, Dean of School

**Step 5 (SCC)**

Open Hearing 10/29/93 Date ☒ Approved by Senate Curriculum Committee 10/15/94 Date

☐ Returned to sponsor(s) for the following reasons:

**Step 6 (Senate)**

Presented to Senate 10/26/94 Date ☒ Approved ☐ Not Approved

Notification to Executive Vice-President/Provost 10/26/94 Date

Ronald J. Galo
Signature SCC Chairperson

## Step 7 (Executive V.P./Provost)

Received 11/17/94
Date

Approved ☒ Yes ☐ No

If no, reasons are as follows:

Student credit hours _____

Faculty load hours _____

Equalized credit hours _____

Official copy and approval sheet filed _____
Date

Donald L. Copeland
Signature, Executive Vice President/Provost

## Registrar

Approved course description received 22 Nov 94
Date

Hegis Taxonomy and Course Number assigned 0501 210

R L McIver
Signature, Registrar

22 Nov 94
Date

## Notification forwarded:

- ☐ Senate Curriculum Committee Chairperson
- ☐ Department Chairperson(s)
- ☐ Academic Dean(s)
- ☐ Registrar
- ☐ Sponsor(s)

Rev 8/88 8007

92-93-C7

**Rowan College of New Jersey**
**Department of Computer Science**

**Course Proposal**

**Foundations of Computer Science** 0707.210

1. Details

| | | |
|---|---|---|
| a. | Course Title: | Foundations of Computer Science |
| b. | Sponsor: | Nancy Tinkham, Department of Computer Science |
| c. | Credit Hours: | 3 |
| d. | Course Level: | Sophomore |
| e. | Curricular Effect: | Required course for computer science majors; Prerequisite for Compiler Design (0704.401), Theory of Computing (0707.422), and Design and Analysis of Algorithms |
| f. | Prerequisites: | Pre-calculus (1701.122) and Discrete Mathematics (1703.150) |
| | Corequisite (may be taken before or at the same time as Foundations of Computer Science): | Data Structures (0704.222) |
| g. | Suggested Time, Implementation | One or two sections per year, beginning as soon as possible. |
| h. | Resources: | Faculty, equipment, and library resources are adequate |

2. Rationale

The rationale for this course is twofold: to reduce duplication of material in the computer science curriculum, and to strengthen the theoretical component of the computer science program.

Some of the foundational material in computer science — in particular, automata theory, context-free grammars, and Turing machines — is now being taught in several different courses: Compiler Design, Theory of Computing, and occasionally Data Structures and Programming Languages. This new course will allow foundational material to be taught in one course, which makes the curriculum more efficient and helps the department to ensure that all the necessary foundational material is taught to students.

Further, comparison of our program with the criteria of the ACM/IEEE-CS Joint Curriculum Task Force indicates that our program is somewhat weak in its coverage of theoretical foundations. A required course introducing students to important foundational material will strengthen our curriculum in this area.

Foundations of Computer Science will be a prerequisite to Compiler Design (which uses grammars and finite automata), Design and Analysis of Algorithms (which uses Turing Machines), and Theory of Computing (which will build on most of the material in the Foundations course).

3. Essence of the course

a. Objectives in relation to student outcome

At the end of this course, students should be able to construct finite automata, regular expressions, grammars, push-down automata, and Turing machines to recognize languages. They should also be able to evaluate propositional and first-order logic expressions, and to write simple proofs using these formalisms.

b. Topic outline

The course will cover the following topics at an introductory level:

Finite automata, regular expressions, and regular grammars
Context-free grammars and push-down automata
Turing machines
Propositional logic
Predicate logic
Program correctness
Applications of the above topics to computer science problems

c. Evaluation of students

Students will be graded on the basis of homework and exams.

d. Course evaluation

The course will be evaluated as part of our department self-study every 5 years.

4. Consultants

Richard Beigel, Computer Science Department, Yale University
John F. Dooley, Motorola Urbana Design Center
Jerry Grossman, Computer Science Department, Oakland University
Jeff McConnell, Chair, Computer Science Department, Canisius College
Bernard Moret, Computer Science Department, University of New Mexico
Hal Perkins, Computer Science Department, Cornell University
James Peterson, Computer Science Department, Colorado State University
John Shepherd, Computer Science Department, Melbourne University
D. E. Stevenson, Computer Science Department, Clemson University
Jan van de Snepscheut, Computer Science Department, California Institute of Technology

Results of consultation:

Courses similar to our proposed sophomore-level Foundations course are taught at Yale University, Oakland University, Canisius College, Melbourne University, and California Institute of Technology. Junior-level Foundations courses are taught at Clemson University and Colorado State University. Most of the consultants from these schools reported that their Foundations course was a good addition to the curriculum, providing students with an early introduction to theoretical concepts and giving them an overview of some of the non-programming aspects of computer science. Several consultants recommended that students take Discrete Mathematics before taking the Foundations course so that they will be adequately prepared for the mathematical content of the course.

**Catalogue description:**

Foundations of Computer Science

*(Prerequisites: Pre-calculus (1701.122), Discrete Math (1703.150); Corequisite: Data Structures (0704.222) )*

This course provides an introduction to the theoretical foundations of computer science, including finite automata, context-free grammars, Turing machines, and formal logic.

Suggested hegis #: 0707.250

Return-Path: beigel-richard@CS.YALE.EDU
Received: from BEECH.THEORY.CS.YALE.EDU by thailand.CS.YALE.EDU via SMTP; Fri, 2
Received: by BEECH.THEORY.CS.YALE.EDU (Sendmail-5.65c/res.client.cf-3.5)
        id AA27401; Fri, 23 Oct 1992 11:45:43 -0400
   :e: Fri, 23 Oct 1992 11:45:43 -0400
From: beigel-richard@CS.YALE.EDU (Richard Beigel)
Message-Id: <199210231545.AA27401@BEECH.THEORY.CS.YALE.EDU>
To: nlt@gboro.glassboro.edu
Cc: beigel@CS.YALE.EDU
Subject: Re: Sophomore-level theory course
Newsgroups: comp.theory,comp.edu
In-Reply-To: <1992Oct23.024658.5591@gboro.glassboro.edu>
Organization: Yale University Computer Science Dept., New Haven, CT 06520-2158
Cc:

Dear Nancy:

I think that most departments require a one-semester discrete math course as a prerequisite for their computability course. Otherwise, you will be repeatedly surprised by the really basic things your students don't know. I like a computability course that covers finite machines and regular expressions, stack machines and context-free grammars, Turing machines and decidability, and NP-completeness. I cover CYK and Earley's algorithm in class, but I leave the rest of parsing theory like LR(k) grammars for a compilers class.

I think you will be interested in my computability textbook, which will be available next year. It presents about the same material as [illegible]ocroft-Ullman (HU), but in a much more unified way. It is also [illegible]itten expressly for computer scientists (whereas HU is written for mathematicians). I've used the manuscript successfully in classes that were about 50% sophomores. I'm at Yale, but the manuscript also has favorable reviews from schools like U. Manitoba.

The textbook, "The Langague of Machines" will be in print next spring. If you would like to see the manuscript when planning your curriculum, you can write to the publisher for a free copy (tell him Richard Beigel sent you). Publisher's address:

Burt Gabriel
Computer Science Press
W.H. Freeman and Company
41 Madison Ave., Floor 37
New York, NY 10010

Sincerely,
Richard Beigel
--

Richard Beigel
Dept. of Computer Science
P.O. Box 2158, Yale Station
New Haven, CT 06520-2158

telephone: (203)432-1228
email: beigel-richard@cs.yale.edu
campus mail: A. K. Watson Hall
51 Prospect Street

Return-Path: jdooley@urbana.mcd.mot.com
Received: from pacific.urbana.mcd.mot.com by urbana.mcd.mot.com (5.61/1.34)
        id AA11619; Wed, 28 Oct 92 13:00:50 -0600
Received: by pacific.urbana.mcd.mot.com (5.61/1.34)
        id AA12839; Wed, 28 Oct 92 13:00:17 -0600
Date: Wed, 28 Oct 92 13:00:17 -0600
From: jdooley@urbana.mcd.mot.com (John Dooley)
Message-Id: <9210281900.AA12839@pacific.urbana.mcd.mot.com>
To: nlt@gboro.glassboro.edu
Subject: Re: Sophomore-level theory course
Newsgroups: comp.theory,comp.edu
References: <1992Oct23.024658.5591@gboro.glassboro.edu>

In comp.edu you write:

>     Our computer science department is planning to rearrange the theory
>part of our curriculum.  In particular, the basics of automata theory,
>grammars, and Turing machines are currently being taught in both our
>compiler design class and our theory of computing class, with the result
>that neither class is able to cover as much advanced material as we
>would like.

>     As a solution, we are planning to introduce a new course, called
>something like "Foundations of Computer Science", at the sophomore level,
>which would introduce automata, grammars, and TMs, possibly along with some
>other material such as propositional and predicate logic.  This material
>could then be removed from the other courses.

>     Has anyone else set up a similar sophomore-level "foundations" course
>in their curriculum?  If so, how well is it working?  This arrangement of
>courses is different from the one at most schools, I think, but it sounds
>like an improvement to me; if anyone else has some empirical data, we could
>benefit from your advice.

>     For those who have such a course, what textbook are you using?
>I've seen Aho & Ullman's _Foundations_of_Computer_Science_, which is the
>best-looking text so far, but it's a thick book that covers more
>material than we could reasonably include in a semester.  Are there other
>books available that cover theoretical topics in a way that's accessible
>to sophomores?  (I'd need something at a more introductory level than
>the standard theory books like Hopcroft & Ullman or Lewis & Papadimitriou.)

>_____

>                                                    Nancy Tinkham
>                                                    nlt@gboro.glassboro.edu

The small college where I teach part-time doesn't have a course
like you describe, but they've been thinking of one and I would be
grateful for any information/pointers/texts that you come up with.

I've taught "foundations" topics in discrete math courses and in an
"Organization of Programming Languages" course and have found both
unsatisfying since I can't get through more than cursory explanations
of topics.

Thanks and good luck.

--
John F. Dooley                    jdooley@urbana.mcd.mot.com

(217) 384-8591 ...uiucuxc!udc!jdooley
Motorola Urbana Design Center, 1101 E. University Ave. Urbana, IL 61801

Article 3928 of comp.theory:
Newsgroups: comp.theory
Path: glassboro.edu!jvnc.net!darwin.sura.net!zaphod.mps.ohio-state.edu!caen!dest
From: grossman@vela.acs.oakland.edu (Jerry Grossman)
Subject: Re: Sophomore-level theory course
Message-ID: <1992Oct23.172506.27778@vela.acs.oakland.edu>
Organization: Oakland University, Rochester MI.
Date: Fri, 23 Oct 1992 17:25:06 GMT
Lines: 18

I've taught such a course using Brookshear, Theory of Computation: Formal Languages, Automata, and Complexity (Benjamin/Cummings, 1989). It went pretty well. The prerequisites were a year of CS (through some data structures) and a Discrete Mathematics course (see below). The book is quite elementary, nowhere near the sophistication level of Lewis and Papadimitriou.

Ideally, students should already have had some exposure to logic, recursion, mathematical induction, graphs, etc. These are some of the topics in a Discrete Mathematics course usually taught in the freshman or sophomore years. (I think it is best to have this after a first programming course and after a year of calculus, to develop CS and mathematical maturity.) Teach the Discrete Math class first semester of sophomore year and your theory course the second semester. Good books for the former would be Rosen's Discrete Mathematics and Its Applications (McGraw-Hill) or Grossman's Discrete Mathematics: An Introduction to Concepts, Methods, and Applications (Macmillan).

Return-Path: mcconnel@canisius.edu
Received: by canisius.edu (4.1/CCCS-2.04)
        id AA22884; Fri, 23 Oct 92 11:48:08 EDT
Date: Fri, 23 Oct 92 11:48:08 EDT
From: mcconnel@canisius.edu (Jeff Mcconnell)
Message-Id: <9210231548.AA22884@canisius.edu>
To: nlt@gboro.glassboro.edu
Subject: Re: Sophomore-level theory course
In-Reply-To: your article <1992Oct23.024658.5591@gboro.glassboro.edu>
News-Path: ub!zaphod.mps.ohio-state.edu!darwin.sura.net!jvnc.net!glassboro.edu!n

We have just put in a second year theory of comp class, which i'm teachi right now. My expectations have changed, even though i'm only in the middle of course. I was hoping to to some of the theoretical stuff, but i'm finding that students can only really handle practical algorithmic concepts (ie: no proofs, u by construction). I'm still really happy with the book: An introduction to form languages and automata by Peter Linz ( DC Heath, 1991?).

Good luck,

Jeff McConnell
Department Chair

Return-Path: moret@chaco.cs.unm.edu
Received: from chaco.cs.unm.edu by unmvax.cs.unm.edu (5.61/3.3) with SMTP
        id <AA26693@unmvax.cs.unm.edu>; Fri, 23 Oct 92 09:23:03 -0600
Received: by chaco.cs.unm.edu (5.57/Ultrix3.0-C)
        id AA23926; Fri, 23 Oct 92 09:23:02 -0600
Date: Fri, 23 Oct 92 09:23:02 -0600
From: moret@chaco.cs.unm.edu (Bernard Moret)
Message-Id: <9210231523.AA23926@chaco.cs.unm.edu>
To: nlt@gboro.glassboro.edu (N. L. Tinkham)
Subject: Re: Sophomore-level theory course

Have you looked at the text of Mandrioli and Ghezzi, "Theoretical Foundations of
than the rather disconnected and rambling monster of Aho and Ullman (which seems
very good taken alone), and clearly more accessible than the Lewis and Papadimit

Savitch was working a couple of years ago on a revision of his undergraduate tex
(whereas the important part of CS theory these days is complexity theory).
The new edition (if and when it comes out) may be a very good choice: Savitch wr

Bernard M.E. Moret                    Department of Computer Science
(505) 277-3112     University of New Mexico, Albuquerque, NM 87131-1386

Return-Path: hal@cs.cornell.edu
Received: from CLOYD.CS.CORNELL.EDU by thialfi.cs.cornell.edu (5.67/I-1.99C)
        id AA11154; Fri, 23 Oct 92 13:38:12 -0400
Received: from AMON.CS.CORNELL.EDU by cloyd.cs.cornell.edu (5.67/I-1.99D)
        id AA02745; Fri, 23 Oct 92 13:38:16 -0400
From: hal@cs.cornell.edu (Hal Perkins)
Date: Fri, 23 Oct 92 13:38:09 -0400
Message-Id: <9210231738.AA22241@amon.cs.cornell.edu>
Received: by amon.cs.cornell.edu (5.67/N-0.13)
        id AA22241; Fri, 23 Oct 92 13:38:09 -0400
To: nlt@gboro.glassboro.edu
Subject: Re: Sophomore-level theory course
Newsgroups: comp.theory,comp.edu
References: <1992Oct23.024658.5591@gboro.glassboro.edu>

In comp.theory you write:
>      For those who have such a course, what textbook are you using?
>I've seen Aho & Ullman's _Foundations_of_Computer_Science_, which is the
>best-looking text so far, but it's a thick book that covers more
>material than we could reasonably include in a semester.  Are there other
>books available that cover theoretical topics in a way that's accessible
>to sophomores?  (I'd need something at a more introductory level than
>the standard theory books like Hopcroft & Ullman or Lewis & Papadimitriou.)

I've looked at the Aho & Ullman book briefly and have one problem with
it.  They cover data structures et al. without any semi-formal
correctness analysis.  Logic is almost an afterthought at the end of
the book.

' ies & Schneider are writing a book that might cover what you want.
. 's intended for a sophomore-level "discrete structures" course and
covers basic logic, graphs, trees, etc.  I don't remember if it covers
the basics of formal language theory (it's been a while since I looked
at the table of contents).  It's being used for our sophomore course
this year and is fairly close to a final draft.  They have circulated
drafts to some colleagues; perhaps they would be willing to send a copy
to you.  For email try: gries@cs.cornell.edu

Hal Perkins                    hal@cs.cornell.edu
Cornell CS

Return-Path: peterson@debussy.CS.ColoState.EDU
Received: from debussy.cs.colostate.edu by ccncsu.ColoState.EDU (5.59/Ultrix2.0-
        id AA25230; Fri, 23 Oct 92 09:59:14 MDT
Received: by debussy.CS.ColoState.EDU (5.57/Ultrix3.0-C)
        id AA07948; Fri, 23 Oct 92 09:56:43 -0600
Date: Fri, 23 Oct 92 09:56:43 -0600
From: peterson@cs.colostate.edu (james peterson)
Message-Id: <9210231556.AA07948@debussy.CS.ColoState.EDU>
To: nlt@gboro.glassboro.edu
Subject: Re: Sophomore-level theory course
Newsgroups: comp.theory,comp.edu
In-Reply-To: <1992Oct23.024658.5591@gboro.glassboro.edu>
Organization: Colorado State University, Computer Science Department
Cc:

How about Brookshear's *Theory of Computation*? In the process of adopting the ACM '91 curriculum, we introduced a required course similar to what you describe covering (most of the) material that we used to cover in an optional senior level automata course. It is, however, a junior level course, having discrete math and data structures as prerequisites. Are your students so well grounded in mathematics that they are ready for calculus and discrete math in their first two semesters in school?

james sends

--

james lee peterson peterson@CS.ColoState.edu
dept. of computer science
colorado state university "Some ignorance is invincible."
fort collins, colorado (voice:303/491-7137; fax:303/491-6639)

Return-Path: jas@cs.mu.OZ.AU
Received: from munta.cs.mu.OZ.AU by mulga.cs.mu.OZ.AU with SMTP (5.64+1.3.1+0.50
        Sat, 24 Oct 1992 16:18:03 +1000 (from jas)
Received: by munta.cs.mu.OZ.AU (920110.SGI/slave-1.1)
        id AA06835; Sat, 24 Oct 92 16:18:00 +1000
Date: Sat, 24 Oct 92 16:18:00 +1000
From: jas@cs.mu.OZ.AU
Message-Id: <9210240618.6835@munta.cs.mu.OZ.AU>
To: nlt@gboro.glassboro.edu
Subject: Re: Sophomore-level theory course
Newsgroups: comp.theory,comp.edu
References: <1992Oct23.024658.5591@gboro.glassboro.edu>

In comp.edu you write:
>      For those who have such a course, what textbook are you using?
>I've seen Aho & Ullman's _Foundations_of_Computer_Science_, which is the
>best-looking text so far, but it's a thick book that covers more
>material than we could reasonably include in a semester.  Are there other
>books available that cover theoretical topics in a way that's accessible
>to sophomores?

Our second course at Melbourne Uni (first is standard Pascal programming)
used the Aho and Ullman book for the first time this year. From what I
heard, it worked very well, although our course has a slightly different
emphasis than yours (it is meant to give a broad view of the field of
Comp Sci, touching on many topics, to give students "the big picture").

Your're dead right that you can't cover it all in one semester (ours are
13 weeks long with 3 lectures and 1 tute per week), but I don't see that
   a problem. As long as it covers all the material you need, use it and
. c any interested students read the other bit.

The lecturer for the course (Andrew Davison, ad@cs.mu.oz.au) can probably
give you more details than me. He constantly extols the virtue of the Aho
and Ullman book.

Cheers, jas

(John Shepherd)

Article 3162 of comp.edu:
Xref: glassboro.edu comp.theory:3927 comp.edu:3162
Newsgroups: comp.theory,comp.edu
Path: glassboro.edu!jvnc.net!darwin.sura.net!gatech!hubcap!steve
From: steve@hubcap.clemson.edu ("Steve" Stevenson)
Subject: Re: Sophomore-level theory course
Message-ID: <1992Oct23.121650.25917@hubcap.clemson.edu>
Organization: Clemson University
References: <1992Oct23.024658.5591@gboro.glassboro.edu>
Distribution: na
Date: Fri, 23 Oct 1992 12:16:50 GMT
Lines: 24

nlt@gboro.glassboro.edu (N. L. Tinkham) writes:

>     As a solution, we are planning to introduce a new course, called
>something like "Foundations of Computer Science", at the sophomore level,
>which would introduce automata, grammars, and TMs, possibly along with some
>other material such as propositional and predicate logic.  This material
>could then be removed from the other courses.

Clemson has had such a course at the *junior* level for two years now. I
am the teacher. We have taken exactly the material you're talking about---
grammars, Tm, lambda-calculus, etc---plus some other background, such as
computability and historical background into a fifteen week course.
We did not find a text that was even remotely what I thought was appropriate.
So I wrote my own.

If you're serious, I can let you have the current dvi file. I found that it
is cheap enough to have the local copy shop copy it (about $15).

--
Steve (really "D. E.") Stevenson  steve@hubcap.clemson.edu
Department of Computer Science,  (803)656-5880.mabell
Clemson University, Clemson, SC 29634-1906

Return-Path: jan@vlsi.cs.caltech.edu
Received: from triton.cs.caltech.edu by vlsi.cs.caltech.edu (4.1/1.34.1)
        id AA14322; Mon, 26 Oct 92 07:49:27 PST
Date: Mon, 26 Oct 92 07:49:27 PST
From: jan@vlsi.cs.caltech.edu (Jan van de Snepscheut)
Message-Id: <9210261549.AA14322@vlsi.cs.caltech.edu>
To: nlt@gboro.glassboro.edu
Subject: sophomore course

Dear Nancy Tinkham,

one of our students sent me a clipping from some Newsgroup about
a new sophomore course.

At Caltech, I have been teaching the sophomore course since the
fall of 1989. until then, there was an introductory programming
course for freshmen, and a number of advanced courses that aim
at late undergraduates and early graduate students. The sophomore
course was set up to bridge the gap. There is no distinction
between a theory and a non-theory course.

The sophomore course, called cs20, is 3-3-3 hours per week
(class-homework-labwork) for three trimesters. The emphasis is
not on a broad overview of all cs subjects, but we pick some
interesting subjects (see list below) and treat them quite
rigorously. We try to get the major ideas across including the
kind of theory that is used to pose and solve problems.

- grammars
- an imperative programming language (Dijkstra's guarded commands,
  semantics, heuristics for program design)
- regular expressions, finite state machines, right-linear grammars
- vlsi: operation of transistor; combinatorial and sequential circuits;
  translation from regular expressions to circuits
- parsing
- halting problem
- predicate calculus
- efficiency of programs (O-notion)
- functional programming, including a LISP interpreter
- program inversion
- concurrent program (with a lot of lab work on multicomputers)
- compilation, including a Pascal-S compiler
- the design of a microprocessor, including such things as
  ALU circuits and microprogram

Most students find the course hard but rewarding. A few don't like
it because it is not your typical hacker's course, but most students
really like it. Since thereis not textbook that covers the topics
of this course, I wrote my own handouts and I am in the process of
polishing them into a draft book to be published by Springer. Let me
know if I can be of any further help,

--Jan van de Snepscheut