

Approval Form

0704.564

Proposal Title: COMPUTER DESIGN THEORY

Sponsor(s) K. AMER Dept.: MATH/COMPUTER SCIENCE Ext. 6044

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

<p>Step 1 (Department)</p> <p><input checked="" type="checkbox"/> Approved <u>10/19/87</u> <small>Date</small></p> <p><input type="checkbox"/> Not Approved</p> <p><u>Ronald J. Gocher</u> <small>Dept. Chairperson</small></p> <p><input type="checkbox"/> Reviewed <u>2.5.1987</u> <small>Date</small></p> <p><u>11-4-87</u> <small>Dept. Chairperson</small></p>	<p>Step 2 (Receipt)</p> <p><input type="checkbox"/> SCC# <u>87-88-25</u></p> <p>Proposal Received <u>11/18/87</u> <small>Date</small></p> <p><u>Brenda A. Bolz</u> <small>SCC Chairperson</small></p>	<p>Step 3 (School CC)</p> <p>Reviewed <u>1/30/89</u></p> <p><input checked="" type="checkbox"/> Approved <input type="checkbox"/> Not Approved</p> <p>Comments:</p> <p><u>Ronald J. Gocher</u> <small>School Curr. Comm. Chairperson</small></p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Step 4 (Academic Dean) **Comments:**

Recommend
 Not Recommend
 Conditionally Recommend (see comments)

Reviewed 2-10-88
Date

McDonnell
Signature, Dean of School

Step 5 (SCC)

Open Hearing 3/14/89 Approved by Senate Curriculum Committee 3/16/89
Date Date

Returned to sponsor(s) for the following reasons:

Curricular Effect Statement ^{OK}
Catalog desc. - Pre-req. clarified
Capitalization -

Step 6 (Senate)

Presented to Senate 3/17/89 Approved Not Approved
Date

Notification to Vice-President for Academic Affairs 3/31/89 Brenda A. Bolz
Date Signature, SCC Chairperson

Step 6 (Senate)

Received 4/3/89
Date

Approved YES No


If no, reasons are as follows:

Student credit hours 3

Faculty load hours 3

Equalized credit hours 3

Official copy and approval sheet filed 4/2/89
Date



Signature, Vice-President for Academic Affairs

Registrar

Approved course description received _____
Date

Hegis Taxonomy and Course Number assigned 0704-564 Rec KRC 4/11/89

Signature, Registrar

Date

Notification forwarded:

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

COURSE PROPOSAL

COMPILER DESIGN THEORY

1. Details

- a. Course Title: Compiler Design Theory
- b. Sponsor: K. Amer
- c. Course Level: Graduate. Three credit hours.
- d. Curricular Effect: Major requirement for the Master's Degree in Junior College Teaching (Computer Science).
- e. Prerequisite: Data Structures and Programming languages.
- f. One section every academic year will be offered.
- g. There is an adequate number of faculty members who are well prepared to teach this course. Computer resource are also adequate. Supplementary texts are available in the library.

2. Rationale

Gaining a deep understanding of Computer Science is a great asset to Junior College teachers even though they are likely to be involved in teaching only relatively elementary classes. Without this understanding they will fail to communicate to their students the important aspects of the material covered in Junior College. A teacher who is only one step ahead of his students is not a good one.

This course is an ideal one for achieving the above goal. It covers many aspects of theoretical computer science. The gained theoretical knowledge is then applied to designing compilers which are sophisticated computer programs whose function is translating high level code to machine language. This course ties theory to applications in a very effective way.

3. Essence

- a. The objectives of the course are:
 - I. The students will study the rationale for compilers. They will also learn different compiler models.
 - II. The students will study finite state machines. They will also apply them to the design of the lexical box of a compiler.
 - III. The students will study pushdown machines, context free grammars, and translation grammars. The students will apply these notions to the design of the syntax box of a compiler.
 - IV. The students will design a code generator for a compiler.

b. Topical Outline

I. Introduction

- A. Rationale for Compilers
- B. Models of Compilers

II. The Lexical box.

- A. Finite State Machines
- B. Implementation of Finite State Machines
- C. The design of a lexical box.

III. The Syntax Box

- A. Pushdown Machines
- B. Context Free Grammars
- C. Translation Grammars
- D. The design of a syntax box.

IV. The Code Generator

- A. The simulation of run time.
- B. Memory layout.
- C. Table entries.

c. Two written exams and three computer projects will be used to evaluate the students. The first two computer projects are: designing a lexical box and a syntax box. The third one, which is not required of the undergraduate students, is concerned with the development of a code generator.

d. Course Evaluation: we plan to evaluate the course by means of student evaluations.

4. Consultation: Richard Smith, Educational Administration

5. Additional Information

A textbook can be chosen from the following list:

- I. Aho, A. V. and Ullman, J.D., Principles of Compiler Design, Theory, Addison-Wesley, 1976.
- II. Lewis, P.M., Rosenkrantz, D.J. and Stearns, R.G., Compiler Design Theory, Addison Wesley, 1976.
- III. Barret and Couch, Compiler Construction, Theory and Practice, SRA, 1979

6. Catalog Description:

GXXXX

Compiler Design Theory (Prerequisite: GXXXXX and GXXXXX).

This course covers the following topics: Compiler models, finite state machines, the lexical box, context free grammars, translation grammars, pushdown machines, the syntax box, the code generator.

COURSE PROPOSAL

COMPILER DESIGN THEORY

1. Details

- a. Course Title: Compiler Design Theory
- b. Sponsor: K. Amer
- c. Course Level: Graduate. Three credit hours.
- d. Curricular Effect: Major requirement for the Master's Degree in Junior College Teaching (Computer Science).
- e. Prerequisite: Data Structures and Programming languages.
- f. One section every academic year will be offered.
- g. There is an adequate number of faculty members who are well prepared to teach this course. Computer resource are also adequate. Supplementary texts are available in the library.

2. Rationale

Gaining a deep understanding of Computer Science is a great asset to Junior College teachers even though they are likely to be involved in teaching only relatively elementary classes. Without this understanding they will fail to communicate to their students the important aspects of the material covered in Junior College. A teacher who is only one step ahead of his students is not a good one.

This course is an ideal one for achieving the above goal. It covers many aspects of theoretical computer science. The gained theoretical knowledge is then applied to designing compilers which are sophisticated computer programs whose function is translating high level code to machine language. This course ties theory to applications in a very effective way.

3. Essence

- a. The objectives of the course are:
 - I. The students will study the rationale for compilers. They will also learn different compiler models.
 - II. The students will study finite state machines. They will also apply them to the design of the lexical box of a compiler.
 - III. The students will study pushdown machines, context free grammars, and translation grammars. The students will apply these notions to the design of the syntax box of a compiler.
 - IV. The students will design a code generator for a compiler.

b. Topical Outline

I. Introduction

- A. Rationale for Compilers
- B. Models of Compilers

II. The Lexical box.

- A. Finite State Machines
- B. Implementation of Finite State Machines
- C. The design of a lexical box.

III. The Syntax Box

- A. Pushdown Machines
- B. Context Free Grammars
- C. Translation Grammars
- D. The design of a syntax box.

IV. The Code Generator

- A. The simulation of run time.
- B. Memory layout.
- C. Table entries.

c. Two written exams and three computer projects will be used to evaluate the students. The first two computer projects are: designing a lexical box and a syntax box. The third one, which is not required of the undergraduate students, is concerned with the development of a code generator.

d. Course Evaluation: we plan to evaluate the course by means of student evaluations.

4. Consultation: Richard Smith, Educational Administration

5. Additional Information

A textbook can be chosen from the following list:

- I. Aho, A. V. and Ullman, J.D., Principles of Compiler Design, Theory, Addison-Wesley, 1976.
- II. Lewis, P.M., Rosenkrantz, D.J. and Stearns, R.G., Compiler Design Theory, Addison Wesley, 1976.
- III. Barret and Couch, Compiler Construction, Theory and Practice, SRA, 1979

6. Catalog Description:

GXXXX

Compiler Design Theory (Prerequisite: ~~GXXXX~~^{0704.322} and ~~GXXXX~~^{0704.204}).

This course covers the following topics: Compiler models, finite state machines, the lexical box, context free grammars, translation grammars, pushdown machines, the syntax box, the code generator.