

CSC 135: COMPUTABILITY AND FORMAL LANGUAGES

In Workflow

1. CSC Committee Chair (shaverdian@csus.edu;%20jouyang@csus.edu)
2. CSC Chair (faroughi@csus.edu)
3. ECS College Committee Chair (figgess@csus.edu)
4. ECS Dean (kevan@csus.edu)
5. Academic Services (torsetj@csus.edu;%20cnewsome@skymail.csus.edu)
6. Senate Curriculum Subcommittee Chair (curriculum@csus.edu)
7. Dean of Undergraduate (james.german@csus.edu;%20celena.showers@csus.edu)
8. Dean of Graduate (cnewsome@skymail.csus.edu)
9. Catalog Editor (torsetj@csus.edu)
10. Registrar's Office (w lindsey@csus.edu)
11. PeopleSoft (PeopleSoft@csus.edu)

Approval Path

1. Wed, 30 Sep 2020 21:31:32 GMT
Anna Baynes (shaverdian): Approved for CSC Committee Chair
2. Wed, 30 Sep 2020 21:36:37 GMT
Nikrouz Faroughi (faroughi): Approved for CSC Chair
3. Mon, 12 Oct 2020 21:52:16 GMT
Gareth Figgess (figgess): Rollback to CSC Committee Chair for ECS College Committee Chair
4. Mon, 12 Oct 2020 22:33:55 GMT
Anna Baynes (shaverdian): Approved for CSC Committee Chair
5. Mon, 12 Oct 2020 23:24:46 GMT
Nikrouz Faroughi (faroughi): Rollback to CSC Committee Chair for CSC Chair
6. Tue, 13 Oct 2020 05:00:15 GMT
Anna Baynes (shaverdian): Approved for CSC Committee Chair
7. Tue, 13 Oct 2020 15:31:28 GMT
Nikrouz Faroughi (faroughi): Approved for CSC Chair
8. Fri, 16 Oct 2020 17:15:17 GMT
Gareth Figgess (figgess): Approved for ECS College Committee Chair
9. Fri, 16 Oct 2020 17:28:13 GMT
Kevan Shafizadeh (kevan): Approved for ECS Dean

Date Submitted: Wed, 30 Sep 2020 19:58:51 GMT

Viewing: CSC 135 : Computability and Formal Languages

Last edit: Mon, 12 Oct 2020 23:24:45 GMT

Changes proposed by: Ted Krovetz (101058577)

Contact(s):

Name (First Last)	Email	Phone 999-999-9999
Ted Krovetz	tdk@csus.edu	000-000-0000

Catalog Title:

Computability and Formal Languages

Class Schedule Title:

Computability and Formal Lang

Academic Group: (College)

ECS - Engineering & Computer Science

Academic Organization: (Department)

Computer Science

Will this course be offered through the College of Continuing Education (CCE)?

No

Catalog Year Effective:

Fall 2021 (2021/2022 Catalog)

Subject Area: (prefix)

CSC - Computer Science

Catalog Number: (course number)

135

Course ID: (For administrative use only.)

201850

Units:

3

In what term(s) will this course typically be offered?

Fall, Spring

Does this course require a room for its final exam?

Yes, final exam requires a room

Does this course replace an existing experimental course?

No

This course complies with the credit hour policy:

Yes

Justification for course proposal:

1. The prerequisite change is to avoid students worried about failing their current course from occupying enrollment. Our current course waitlists are filled.
2. The Computer Science department reviewed our courses based on current teaching practice and professional organization recommendations. This update is required for Computer Science program external accreditation.

Course Description: (Not to exceed 80 words and language should conform to catalog copy.)

Introduction to limits of computation and techniques for specifying and processing formal languages. Regular languages, regular expressions, finite automata, properties and limitations. Context-free languages, grammars, pushdown automata, properties and limitations. Applications in lexical and syntax analyses, including recursive-descent or table-driven parsing. Turing machines, halting problem, reductions. Introduction to functional programming and related programming language features.

Are one or more field trips required with this course?

No

Fee Course?

No

Is this course designated as Service Learning?

No

Does this course require safety training?

No

Does this course require personal protective equipment (PPE)?

No

Does this course have prerequisites?

Yes

Prerequisite:

CSC 28, CSC 35, CSC 130, and not currently enrolled in CSC 135.

Prerequisites Enforced at Registration?

Yes

Does this course have corequisites?

No

Graded:

Letter

Approval required for enrollment?

No Approval Required

Course Component(s) and Classification(s):

Lecture

Lecture Classification

CS#02 - Lecture/Discussion (K-factor=1WTU per unit)

Lecture Units

3

Is this a paired course?

No

Is this course crosslisted?

No

Can this course be repeated for credit?

No

Can the course be taken for credit more than once during the same term?

No

Description of the Expected Learning Outcomes: Describe outcomes using the following format: 'Students will be able to: 1), 2), etc.'

Students completing this course will be able to

1. Design generators and recognizers for simple regular and context-free languages and give examples of algorithmically converting from one to the other;
2. Demonstrate a context-free grammar ambiguous using parse trees;
3. Explain the limits of regular and context-free languages and apply a pumping lemma to show a language not to be regular and/or context-free;
4. Discuss parsing and scanning techniques including the principle of longest substring, recursive-descent or table-driven parsing, derivations, parse trees, first and follow sets;
5. List the main phases of a compiler, and the function of each;
6. Explain functional programming language features and their use, including immutability, higher-order functions, lambda functions, dynamic type systems, and tail recursion;
7. Develop solutions to simple problems using a functional programming language; and
8. Describe the importance of Turing machines, the halting problem, and reductions that prove certain problems cannot be solved by computer algorithms.

Attach a list of the required/recommended course readings and activities:

135_23SEP2020.docx

Assessment Strategies: A description of the assessment strategies (e.g., portfolios, examinations, performances, pre-and post-tests, conferences with students, student papers) which will be used by the instructor to determine the extent to which students have achieved the learning outcomes noted above.

Outcomes 1-8 are assessed via regular quizzes, exams, and a final exam. Outcome 7 is also assessed via graded programming assignments.

Is this course required in a degree program (major, minor, graduate degree, certificate?)

Yes

Has a corresponding Program Change been submitted to Workflow?

No

Identify the program(s) in which this course is required:

Programs:

BS in Computer Science

Does the proposed change or addition cause a significant increase in the use of College or University resources (lab room, computer)?

No

Will there be any departments affected by this proposed course?

No

I/we as the author(s) of this course proposal agree to provide a new or updated accessibility checklist to the Dean's office prior to the semester when this course is taught utilizing the changes proposed here.

I/we agree

University Learning Goals

Undergraduate Learning Goals:

Intellectual and practical skills

Is this course required as part of a teaching credential program, a single subject, or multiple subject waiver program (e.g., Liberal Studies, Biology) or other school personnel preparation program (e.g., School of Nursing)?

No

GE Course and GE Goal(s)

Is this a General Education (GE) course or is it being considered for GE?

No

Reviewer Comments:

Ted Krovetz (tdk) (Wed, 30 Sep 2020 20:00:18 GMT): Note: CSC 135 was previously double-counted as GE B5 to maintain the ability of students to graduate within 120 units. The adjustment to the math and science requirements in this proposal frees 3 units, allowing us to rescind CSC 135's GE B5 status and for students to continue graduating with 120 units. The CSC GE B5 rescinding is in progress.

Ted Krovetz (tdk) (Wed, 30 Sep 2020 20:03:01 GMT): The CSC department has submitted a Form B that reduces the number of required units for the BS by three units. Those three units are what allows CSC 135 to no longer be considered GE B5 and to allow students to graduate with 120 units.

Gareth Figgess (figgess) (Mon, 12 Oct 2020 21:52:16 GMT): Rollback: As requested

Nikrouz Faroughi (faroughi) (Mon, 12 Oct 2020 23:24:46 GMT): Rollback: update

Key: 1038