IC-UNICAMP

Courses from Institute of Computing

MC921 2s20

ATENÇÃO

Aula introdutória será ao vivo na Qua. 16/09 às 20h00. Favor ver link no calendário da disciplina ou aqui (http://meet.google.com/hzp-ktiz-kyg).

Links

- Agenda (https://docs.google.com/spreadsheets/d/1yculItZMIjpmQdZoO6D_SlgMkjquGvIdL62GBzOBN88/edit#gid=0)
- Slides (https://drive.google.com/drive/folders/19JbJpWwoExw0hSi2rU_JcwtD_-NMNx2O?usp=sharing)
- Videos (https://drive.google.com/drive/u/0/folders/1d-OpuDhhZ20Wlh_c4QlwFzJhxExKQFyC)
- Meet classes (http://meet.google.com/hzp-ktiz-kyg)
- Bulletin board (https://classroom.google.com/u/0/c/MTQ4NTQwODU2NzA4)
- Grades (https://drive.google.com/file/d/1SQxnt3JylYimGDJKprtYPSFEgN7m5COq/view?usp=sharing)

Adm

- Theory sessions: Mon. and Wed. (21h – 22h)
- Lab sessions: Mon. (22h – 23h) and Wed. (20h -21h)
- Professor: Guido Araujo
- TA: Caio Salvador

Bulletin board

The course has a bulletin board (https://classroom.google.com/u/0/c/MTQ4NTQwODU2NzA4) for announcements and posts about the course progress. Students are required to closely follow the messages posted on the board, as they include very relevant courseware information.

Syllabus

Classes will use a set of slides (https://drive.google.com/drive/folders/19JbJpWwoExw0hSi2rU_JcwtD_-NMNx2O?usp=sharing) and videos (https://drive.google.com/drive/u/0/folders/1d-OpuDhhZ20Wlh_c4QlwFzJhxExKQFyC), available through the course agenda (https://docs.google.com/spreadsheets/d/1yculItZMIjpmQdZoO6D_SlgMkjquGvIdL62GBzOBN88/edit#gid=0). If necessary, additional lecture notes as well as articles discussed in class will be made available.

Slides and videos are intellectual property of the books' authors, instructors or UNICAMP, and cannot be distributed without their previous authorization.

The course will use material from the following books:

- Andrew Appel. Modern Compiler Implementation in Java (http://www.amazon.com/Modern-Compiler-Implementation-Andrew-Appel/dp/052182060X/ref=sr_1_16?ie=UTF8&qid=1411044376&sr=8-16&keywords=compilers).
- Aho, Sethi and Ullman. Compilers: Principles, techniques and tools (http://www.amazon.com/Compilers-Principles-Techniques-Alfred-Aho/dp/0201100886/ref=sr_1_4?ie=UTF8&qid=1411044323&sr=8-4&keywords=compilers).
- Keith Cooper and Linda Torczon. Engineering a Compiler (http://www.amazon.com/Engineering-Compiler-Second-Edition-Cooper/dp/012088478X/ref=sr_1_2?ie=UTF8&qid=1411044280&sr=8-2&keywords=compilers).

Assignments

The final course grade will be based on 7 programming projects, and 3 exams distributed as follows:

- Projects: 70% (10% each)
- Exams: 30% (10% each)

Projects will use the GitHub Classroom environment, where each project has an associated template repository. Students have to pull the assignment template locally to work, and push it for testing, and before the deadline for grading. The GitHub system will run the tests and automatically compute the assignment grade.

All test inputs for the projects are open, and there are no closed tests. The correct output for each test is open, and their evaluation will take into consideration not only execution correctness but also performance.

GitHub will automatically close the submission system after each project deadline, and there will be **no extensions**. Hence, we **strongly** recommend that the student submit its work even if the testing is incomplete.

A link to each project notebook, containing a detailed description of the project, programming guidelines, code snippets, etc. can be found in the appropriate entry of the course agenda. (https://docs.google.com/spreadsheets/d/1yculItZMIjpmQdZoO6D_SlgMkjquGvIdL62GBzOBN88/edit#gid=0)

Grades

Grades (https://drive.google.com/file/d/1SQxnt3JylYimGDJKprtYPSFEgN7m5COq/view?usp=sharing) will be available at most 15 days after the project/exam due date.

Regarding the calculation of the course final grade, the following rules apply:

- The average of the exams will be computed as EA = sum(Ei*0.10, i = 1-3).
- The average of the projects will be computed as PA = sum(Pi*0.10, i = 1-7).
- The average grade of the course is A = EA + PA.
- Students with any Ei < 5.0 (Ei, i = 1-3) will be required to take the Final Exam (F).
- Students with any Pi < 5.0 (Pi, i = 1-7) will be required to take the Final Exam (F).
- The final course grade G = min (5.0, A), if A < 5.0, or (A + F) / 2 otherwise.

Grade review requests must follow the rules below:

- Review requests must be made exclusively through this form (https://docs.google.com/forms/d/15CJwCrOhkCvtvoXmzQNGdk8jFwxxTLYeNaXcc2oWNoY/edit?usp=sharing).
- Review requests will be received only within 48 hours after the grade is released. After that, it will not be considered.
- The review will be done within 10 days after the request is received, and the result will be informed to the student via his/her DAC/Unicamp e-mail.

Collaboration policy

Projects are individual assignments. Students can collaborate with the goals of understanding and discussing the assignment solution. Nevertheless, code sharing and copying are not allowed, and will be considered fraud.

Exams are individual assignments, and collaboration for their execution is not permitted. Any violation will be considered fraud.

Frauds will not be accepted, G = 0.0 will be assigned to everyone involved, and the case will be brought to the Undergraduate Dean.

□

#	Date	Syllabus	Slides	Videos	Exams	Available	Due	Notes
1	21/9	Introduction and tokens	Link	Link		P1		Lexer
2	23/9	Regular expressions	Link	Link				
3	28/9	DFA and NFA conversion	Link	Link				
4	30/9	Parser table, ambiguity and LL(1) parsing	Link	Link		P2	P1	Syntatic analysis and parser
5	5/10	LR(0), LR(1) and SLR parsing	Link	Link				
6	7/10	Ambiguity, precedence and error recovery	Link	Link				
7	12/10	Holiday						
8	14/10	Semantic analysis rules and guideliines (1/2)	Link	Link		P3	P2	Abstract syntax tree
9	19/10	Semantic analysis rules and guideliines (2/2)	Link	Link				
10	22/10	Abstract Syntax Tree (AST)	Link	Link	E1			
11	26/10	Symbol table and semantic analysis	Link	Link				
12	28/10	Holiday				P4	P3	Semantic analysis
13	2/11	Holiday						
14	4/11	IR, DAGs and BBs	Link	Link				
15	9/11	uC IR	Link	Link				
16	11/11	Code generation (1/2)	Link	Link		P5	P4	Code generation
17	16/11	Code generation (2/2)	Link	Link				
18	18/11	Stack-frame	Link	Link				
19	23/11	Instruction selection and local register allocation	Link	Link				
20	25/11	LLVM IR	Link	Link	E2	P6	P5	uC optimization
21	30/11	Data-flow analysis	Link	Link				
22	2/12	Basic optimizations	Link	Link				
23	7/12	Liveness analysis	Link	Link				
24	9/12	Global registar allocation	Link	Link				
25	14/12	Livness analysis	Link	Link				
26	16/12	Global registar allocation	Link	Link		P7	P6	LLVM optimization
27	21/12	Loop optimization	Link	Link				
28	23/12	Compiler technologies and R&D	Link	Link				
29	4/1	No classes	Link	Link				
30	6/1	No classes	Link	Link	E3	E	P7	
29	20/1	Final Exam					E	