University at Albany
College of Engineering and Applied Sciences / Department of Computer Science

ICEN333/ICSI333
Programming at the Hardware-Software Interface
Spring 2017

Instructor: Vladimir Kuperman (vkuperman@albany.edu)

Head Teaching Assistant: Oguz Aranay (oaranay@albany.edu)

Teaching Assistants: Julia Turner (jturner3@albany.edu),

Noah Park (jpark28@albany.edu),

more to be announced

SUMMARY

Computers have changed the world dramatically. Through the decades of using computers they increased their power, performance, and efficiency million times. However, there is something in common between first computers and contemporary computers. Computers are still a hardware that changes its states by following software programs, and these states and programs are still sets of bits finally.

Introduction to assembly language helps to understand how programs run physical devices; C language is a great bridge between programming languages to develop applications and low level programming software.

Basics of the contemporary computers architecture, understanding low level processes and binary data representation, fetch-execute cycle and programming tools to manage all these things are important for computer professionals.

LEARNING OBJECTIVES

After completing this course the student should be able to:

- Understand the fundamental concepts of programming at the hardware-software interface.
- Demonstrate how typical data values in software and hardware, and machine instructions are represented, manipulated, stored, and transformed at the bit level in response to programmed instructions in C and in assembly language.
- Code, test, debug and internally document computer programs in the C language and in assembly language so they follow given functional specifications, within Unix/Linux systems, using appropriate software tools and practices.
- Understand the workings of the fetch-execute cycle, assemblers, function calling and return, register versus random access memory, interrupts, and trap instructions.

PREREQUISITES

Grade of C or better required in I CSI 310. I CSI310 may be taken concurrently with I CSI333 and in some cases waived by students with sufficient background and Departmental Permission.

TEXTS[1]

Required:

1. H. M. Deitel and P. J. Deitel, *C How to Program*, Pearson, 7th or 8th edition.
2. D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Morgan-K/Elsevier, 4th or 5th edition.

Recommended:

1. S. Harbison and G. Steele, ***C: A Reference Manual*** (any edition).

TENTATIVE TOPIC SCHEDULE[2]

The following schedule is preliminary and may be changed as the semester progresses. The final schedule and specific homework and lab assignments and materials will be provided.

Class	Date	Topic	Lec #
1	Tue, Jan 24, 2017	Hierarchy of Data. Number Systems	L1
2	Thu, Jan 26, 2017	Hierarchy of Programming Languages. Basics of C	L2
3	Tue, Jan 31, 2017	A Simple C Program. Compiling and Executing	
4	Thu, Feb 2, 2017	Assignment Statement. Functions	L3
5	Tue, Feb 7, 2017	Strings. Structures	L4
6	Thu, Feb 9, 2017	Pointers. Dynamic Memory Allocation	L5
7	Tue, Feb 14, 2017	Files	L6
8	Thu, Feb 16, 2017	Files	L7
9	Tue, Feb 21, 2017	Strings and Pointer Arithmetic. Functions for Strings	L8
10	Thu, Feb 23, 2017	Pointer Arithmetic. Bitwise Operators	L9
11	Tue, Feb 28, 2017	Pointer Arithmetic. Bitwise Operators	
12	Thu, Mar 2, 2017	The C Preprocessor	L10
13	Tue, Mar 7, 2017	Final Remarks on C	L11
14	Thu, Mar 9, 2017	**Midterm Exam**	
	Tue, Mar 14, 2017	class suspended	
	Thu, Mar 16, 2017	class suspended	
15	Tue, Mar 21, 2017	An Abstract View of a Computer	L12
16	Thu, Mar 23, 2017	An Abstract View of a Computer	

[1] Any electronic version of the texts will be satisfactory for the course learning objectives.
[2] Largely based on course taught by Prof. S. S. Ravi in Fall 2013.

17	Tue, Mar 28, 2017	MIPS Assembly Language (MAL): Part I (Basic Instructions)	L13
18	Thu, Mar 30, 2017	MIPS Assembly Language (MAL): Part I (Basic Instructions)	
19	Tue, Apr 4, 2017	MIPS Assembly Language (MAL): Part II (Procedure Calls in MAL)	L14
20	Thu, Apr 6, 2017	MIPS Assembly Language (MAL): Part II (Procedure Calls in MAL)	
	Tue, Apr 11, 2017	class suspended	
21	Thu, Apr 13, 2017	MIPS Assembly Language (MAL): Part III (Arrays in MAL)	L15
22	Tue, Apr 18, 2017	MIPS Assembly Language (MAL): Part III (Arrays in MAL)	
23	Thu, Apr 20, 2017	MAL: Part IV (MAL and True Assembly Language (TAL))	L16
24	Tue, Apr 25, 2017	MAL: Part IV (MAL and True Assembly Language (TAL))	
25	Thu, Apr 27, 2017	MIPS Assembly Language (MAL): Part V (MIPS Instruction Formats)	L17
26	Tue, May 2, 2017	MIPS Assembly Language (MAL): Part V (MIPS Instruction Formats)	
27	Thu, May 4, 2017	Review	
28		**Final Exam as scheduled by the University**	

CLASS SCHEDULE

This schedule is based on two 80 minute lectures and one 55 minute lab per week. Each student must be registered for the lecture class and one of the lab classes.

Course	Class #	Type	Meeting Time	Location	Instructor / TA	Office Hours
ICSI 333	7563	Lecture	TTH 04:15pm-05:35pm	LC0020	Kuperman	T 05:45pm-07:00pm
ICSI 333	7623	Lab	F 10:25am-11:20am	SL0G12	Kuperman / TBD	TBD
ICSI 333	7624	Lab	F 12:35pm-01:30pm	SL0G12	Kuperman / Park	TBD
ICSI 333	7625	Lab	F 01:40pm-02:35pm	SL0G12	Kuperman / Turner	TBD
ICSI 333	9142	Lab	W 12:35pm-01:30pm	SL0G12	Kuperman / Aranay	TBD
ICEN 333	10345	Lecture	TTH 04:15pm-05:35pm	LC0020	Kuperman	T 05:45pm-07:00pm
ICEN 333	10347	Lab	F 10:25am-11:20am	SL0G12	Kuperman / TBD	TBD
ICEN 333	10348	Lab	F 12:35pm-01:30pm	SL0G12	Kuperman / Park	TBD
ICEN 333	10350	Lab	F 01:40pm-02:35pm	SL0G12	Kuperman / Turner	TBD
ICEN 333	10352	Lab	W 12:35pm-01:30pm	SL0G12	Kuperman / Aranay	TBD

ATTENDANCE

Blackboard will be used to provide essential course materials, the most current syllabus, and assignment submissions and documents. However, this is not an online course and class and lab attendance is essential.

Although attendance in lectures is not required, students are strongly advised to attend the lectures. Attendance will be taken in each lab class. Missing three or more lab classes results in an automatic E grade for the course. If a student misses a lecture or a lab class, it is his/her responsibility to find out the material covered in that lecture or lab class. It will not be possible for professor or lab instructor to conduct makeup classes.

GRADING POLICY AND ASSESSMENT

Quizzes	8%
Midterm Exam	20%
Programming Projects	42%
Final Exam	30%

Grading Scale (rounded to the nearest whole number):

F	D-	D	D+	C-	C	C+	B-	B	B+	A-	A
0-59	60-62	63-66	67-69	70-72	73-76	77-79	80-82	83-86	87-89	90-92	93-100

Quizzes:

Three quizzes which will be given during your lab classes. All quizzes will be closed book/closed notes. For each student, the lowest quiz score will be dropped. The duration of each quiz will be about 20 minutes. Students must note the following.

- You must take your quizzes during the lab class for which you are registered.
- No makeup quizzes will be given.

Programming Projects:

There will be five programming assignments. These assignments will be tested on the Unix machines provided by the Information Technology Services (ITS) unit of the University. You can log on to these machines remotely. Some of the programming assignments must be done in C while others are to be done in assembly language. Some of the assignments may be done as a team.

Makeup Exams:

Makeup exams will be given only for valid and verifiable extenuating circumstances (e.g. a major medical situation). In such a case, it is the student's responsibility to contact instructor ahead of time and arrange to take a makeup exam at an alternate date/time. Makeup exams will be generally harder than the regular exams.

Policy on Cheating:

1) Cheating in an exam or a quiz will result in an E grade for the course. Further, all the students involved will be referred to the Vice Provost for Undergraduate Studies office as specified by University regulations.

2) Borrowing code from another student (or another team in the case of team projects) will be considered cheating. Likewise, any attempt to obtain partial or full source code for a programming assignment from the Internet will also be treated as cheating. Cheating in a programming assignment will result in the following penalty for all the students involved:
   - The programming assignment in which cheating occurred will be assigned a grade of ZERO.
   - Further, the highest score among the other programming assignments will be changed to ZERO.

- Students who cheat in two or more programming assignments will receive an automatic E grade for the course. The names of such students will also be forwarded to the Vice Provost for Undergraduate Studies office as specified by University regulations.

RESPONSIBLE COMPUTING

Students are required to read the University at Albany Policy for the Responsible Use of Information Technology
https://wiki.albany.edu/display/public/askit/Responsible+Use+of+Information+Technology+Policy

Students will be expected to apply the policies discussed in this document to all computing and electronic