

CS 201 Program Design with C

Debzani Deb

Announcement

- Fall 2007 Scholarship opportunities
 - Pick Up application in CS office (EPS 357)
 - Applications are due on Friday, February 9, 2007.
 - Sonderegger, Micron, Zoot, Rightnow, Telesoft Scholarships
 - In addition to applying to the department, you must apply online to COE by February 1, 2007.
 - <http://www.coe.montana.edu/scholarships>
 - Students who neglect to apply to COE, will not be considered.
 - Continuing student must reapply.

Outline

- Contact details
- Who am I?
- Syllabus
- My policy
- Course Description

How to Contact me

- Name: Debzani Deb
- Email: debzani@cs.montana.edu
- Office: 352, EPS building
- Phone: 994-5978 or 994-4780
- WWW: <http://www.cs.montana.edu/~debzani>
- Office Hours:
 - Monday: 1.10-2:00PM, Wednesday 12.10-1:00PM , Friday 11:00-11:50AM
 - By appointment (please call or email to set up a time)
 - In the classroom right after each class.

How to Contact your TA

- Name: M. Muztaba Fuad
- Email: fuad@cs.montana.edu
- Office: 345, EPS building
- Phone: 994-2225 or 994-4780
- WWW: <http://www.cs.montana.edu/~fuad>

Who am I?

- I am a graduate student here in CS dept.
- Originally I am from Bangladesh.
 - A small country in South Asia
- I got my B.Sc from a science and technology university back in Bangladesh.
- I got my M.Sc in computer science from University of Adelaide, Australia.
- Now doing PhD here in MSU.
- My research Interests: Autonomic computing, parallel and distributed systems, distributed data mining.

Syllabus(1)

- **Course number** : CS 201, CRN: 36819, Credits: 3
- **Prerequisite**: CS 160, **Corequisite**: MATH 160
- **Lectures**: Mon & Wed: 3:10-4:00PM at ROBH 101
- **Labs**:
 - Monday 8:00-9:50AM & 10:00-11:50AM at EPS 254
 - Wednesday 1:10-3:00PM at EPS 254
- **Course webpage**:
<http://www.cs.montana.edu/courses/201>
You can access this page from CS website. I will be posting there all course related information including lecture notes (in pdf). Make sure to visit this often.

Syllabus(2)

- **Textbook**: "Problem Solving and Program Design in C", by Jeri R. Hanly & Elliot B. Koffman, Publisher: Addison Wesley; 5th Edition, ISBN:0-321-40991-4
- **Additional Textbook**: "C++ for Java Programmers", by Mark Allen Weiss, Publisher: Prentice Hall; ISBN:013919424X
It is not necessary to buy this book
- You are expected to read all relevant readings before the class. The class notes and the readings will be posted in the website before the class.
- You may bring the textbook during the class, but it is not mandatory.

Syllabus(3)

- **Grading Criteria**:
 - 2 exams during course and daily quizzes: 30%
 - Final exam (covers the whole semester work: lectures, readings, assignments): 20%
 - Labs and homeworks: 50%, don't expect linear distribution in all labs.
 - In case of programming assignments, your program must compile and execute properly to be graded. It is not enough to try hard, you need to be successful too. So start programming assignments well before the deadline and if you face trouble contact your TA/me or get other available helps.

My Policy: Attendance

- **Attendance**:
 - Class attendance is not mandatory. However, anything you miss for not being in the class is your responsibility.
 - Daily quizzes missed due to absence can not be made up.
 - Participation in labs are strongly recommended. Your TA may make it mandatory, so ask him about his policy.

My policy: Grading

- **Grading Policy**:
 - $\geq 90\%$ \rightarrow A
 - $89\% - 80\%$ \rightarrow B
 - $79\% - 70\%$ \rightarrow C
 - $69\% - 60\%$ \rightarrow D
 - $\leq 59\%$ \rightarrow F
- Depending on the overall class performance this may change.

My policy: Labs & Exams

- I expect you to read all handouts.
- No cheating on assignments or class tests. Everybody should develop their own solution. No sharing or modifying someone else's code.
 - Names of the persons involved in **plagiarism** will be reported to the Dean.
 - I will never try to determine who is the original developer and who copied. All parties will be equally treated.
- No make up tests (unless you show me the proof of a medical emergency) or extra credit assignments.
- No late assignments.

My Policy: missing lab & exams

- **Missing lab assignments:** Nothing can be done
- **Missing Exams: Do not miss exams**
 - If you must, then let me know well before the exam with proof of valid reasons, I will either replace your test score with some other test score or will ask for an makeup exam.
 - If you do not inform me before the exam or do not have a valid reason, you will get a zero.
- **Help on Assignments:** You are most welcome to ask help from me/TA about your assignments. But do not wait until last moment. I have specific office hours and your TA should too, please respect those hours.

Laboratory Policy (1)

- All labs will be done using Linux on the PC's or esus.
- Lab assignments are due at the end of the next lab. This may vary, so be sure to check this for each lab.
- You must have an esus account to login to Linux on the machines in EPS 254.
 - There is no lab this week, so make sure you have your account set up by this week.
- You will need to learn basic Linux commands, a Linux-based text editor, such as pico, vim or emacs and eclipse IDE(CDT).
 - See the links section in the course web page for useful links.

Laboratory Policy (2)

- Review the assignment and read the related readings before coming to the lab.
- In all 3 exams, questions will be asked from the related lab assignments. It is strongly recommended that you do all assignments.
- To make sure that your submission fulfills all requirement, show up for your labs and talk to your TA before submitting.
- There is a lot of work to do in CS201. Remember, this is where you build a strong programming foundation, so do not fall behind.

Laboratory Policy(3)

- **Lab Enrichments**
 - These are the things that may interests advanced students and cover: C extra, additional lab procedures and C++ related topics.
 - No exam questions will be based on enrichment topics.
 - No lab grade will be based on these topics.

Course Content

- 75%: programming in C, using standard C libraries (ANSI C)
- 10%: Usage of Linux
- 15%: Introduction to C++.

Course Goals

- This course is designed to provide students with sufficient foundations to create computer applications in the C language.
- Students will learn imperative programming with C and become familiar with C standard library.
- Course covers usage of pointers, different aspects of memory management in C and structures.
- Students will gain hands on experience through different programming assignments.

Imperative Programming

- Imperative (procedural) programming is a programming paradigm that describes computation as statements that change a program state. Example: C, Java, Ada, Pascal, Fortran.
- Imperative programs are a sequence of commands for the computer to perform.
- Declarative languages are often thought of as defining "what" is to be computed, rather than "how" the computation is to take place, as an imperative programming language does. Example Prolog, SQL.

Course Outcome

- At the end of the course, students should be able to:
 - Build computer applications using C language and apply their knowledge to solve real-life applications.
 - Test and debug their C code.
 - Take advantage to major capabilities of C and C++ programming languages.
 - Understand, appreciate and respect power of pointers in C programming.

Please Introduce Yourself

- Fill up both forms as instructed.
- I expect that you will check your emails regularly.