Welcome to Preparatory C Programming Lab!

Revision Date: July 22, 2016

Introduction

C Programming is the prep course for ESD curriculum. Its a Pass (P) or Fail (F) grade course. Topics include, but are not limited to, the basics of programming, debugging, testing, robotics programming, microcontroller programming, programming games and few operating system related programming. Additionally, students will learn about the *Linux* operating system and editing.

Office Hours and Appointments

Dr. Madhav Rao, 112 IIITB, mr@iiitb.ac.in
Office Hours: Tuesday - 9 AM to 10 AM

TA: XXX

XXX will be your TA for this course. You can reach him at this email id: *yyy@iiitb.org.*

Editors

An *editor* is a word processor for programmers and is used by programmers to write and modify *source code.* Source code refers to the words and punctuation that computer programs start out as. With regards to editing, a programmer's primary concerns are entering source code quickly and modifying source code even more quickly. There are two great and timeless editors geared especially for proficient programmers: *emacs* and *vi* (nowadays *vim*). In this class, you will learn *vim* (although you can also choose to learn *emacs* if you wish). To run the *vim* tutorial, type the command:

```
vimtutor
```

at the system prompt. To exit a *vim* session, type `ZZ` to save your changes and quit and `:q` or `:q!` to quit without saving changes.

You will also learn the Linux operating system. Linux is an open-source version of the UNIX operating system. UNIX was designed by programmers for programmers.

Here are some web tutorials:

- vi cheat sheet
- Quick vi tutorial
- linux cheat sheet
- Quick linux-Unix tutorial
- more linux-unix info
- emacs reference card
- quick emacs tutorial

Activities

In-class and out of class activities will be given to students. For an in-class activities, students are supposed to demonstrate each activity to Instructor before leaving the session. Failure to do so, will result in losing points for that activity. The out of class activities will be given to students regularly. Failure to demonstrate atleast one activity will result in Fail grade. During demonstration, Instructor will ask to modify the programs and students are supposed to demonstrate the modified version as well. If the instructor sees that a student is not able to work on his/her own, the instructor will consider this as not his work/development.

Catalog Statement

An preparatory course on programming. Language concepts: datatypes, expressions, variables, control flow, functions, recursion, arrays, pointers, structures, and unions. Software engineering concepts: testing and debugging. Operating system concepts: directories, paths, files, and text editing. Applications: Robotics, microcontroller programming.

Course Summary

This course introduces C, an important system level programming language. Using C, students will learn the basics of programming, including the designing, writing, testing, and debugging of programs. In addition, students will learn basic system tasks for the purposes of organizing and maintaining a suite of C programs.

Course Objectives

At the end of this class, a student should understand the concepts of:

- *defining, using, and modifying variables*
- *formulating expressions to represent desired quantities*
- *controlling the execution of code within a program*

- *defining and calling functions*

- *generating thorough test suites*

- *primitive debugging skills to solve semantic program faults*

- *organizing code using system utilities*

Textbook

The recommended textbooks are

- *The C Programming language* by Kernighan and Ritchie.

- *How to solve it by Computers* by Dromey

- *Code Complete* by McConnell

C Programming lab	
Topic	hours
Introduction to Vim, C Framework and Linux	3
Literals, Variables, Operators, Assignment	3
Command line arguments and File I/O	3
Control Flow	3
Functions	3
Recursion	3
Arrays	3
Pointers	6
Structures	3
Nodes	3
Robotics framework installation and simulation	6
Total Hours	39

Additionally, students are encouraged to visit the following links to understand more on unix enviroment and vim editing skills.

- Text-editing http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Editors/

- Directories and paths http://www.ee.surrey.ac.uk/Teaching/Unix/

- Manipulating files http://www.ee.surrey.ac.uk/Teaching/Unix/

Unexcused absence will result in losing of points for the activity involved in the class. A maximum of two excused absences are allowed. More absences will be reported to the Registrar. Students who have excused absences are supposed to demonstrate the activities which they had failed to show, immediately after they start reporting to class.

Other Details

Please be in your seat by the beginning of class. There is to be no food or drink in the classroom. Sleep is allowed only in the event of a stultifyingly boring lecture. Being caught reading a newspaper, using the computer for purposes other than the class or texting on a cell phone will count as half an unexcused absence (losing 1 point from the overall grade). Schedule changes, assignment clarifications, and such will be announced via the LMS forum.

If your assignment looks as if you have plagiarized someone else's assignment, both assignments will be forwarded to the Director of the IIITB. Further you will be considered guilty of plagiarism. The minimum punishment requested for plagiarism, or any other academic misconduct, is dismissal from the class with a failing grade. You are required to protect your work from plagiarism. If your work is plagiarized, it will be assumed that you were a willing participant in the plagiarism and you will receive the same punishment as the plagiarist, absent evidence to the contrary. Interaction with fellow classmates on programming activities is encouraged, but there should be no sharing of code or solutions, only an exchange of high-level ideas. Do not share your code or email your code to fellow classmates, this can be tracked and you will be punished severely based on this conduct.