CS197c
Intro to C
Nicolas Scarrci
University of Massachusetts Amherst
Contact information

- Office: A331 in LGRC
- Email: nscarrci@cs.umass.edu
- website: http://people.cs.umass.edu/~nscarrci/cs197c
This class is an introduction to programming in C.
- We will NOT discuss C++.

This class assumes knowledge of JAVA.
- The goal is to introduce you to lower level programming concepts not present in JAVA.
This is a 1-credit Pass/Fail course.
- It does not count for the major/minor.

- We meet every week for 6 weeks.
Assignments

- 5 one-week assignments.
  - Due Wednesdays at 11:59 PM

- 5 worksheets.
  - Due Fridays at 11:59PM
Grading

- 5 one-week assignments.
  - 15 points a piece.

- 5 worksheets.
  - 5 points a piece.

- 100 points total.
  - You need 65 to pass.
Your assignments are graded as follows

- 5 points for effort.
  - Comment out code that does not work. It helps us assess how much effort you've put in.

- 10 points for functionality.
  - Each assignment comes with a rubric explaining how the 10 points are distributed.
Grading

- Your worksheets are auto-graded
  - You may work together in class.

- Week long assignments are manually graded.
  - You may not work together, though you may discuss the project.

http://www.umass.edu/ombuds/honesty.php/
Grading

- Your assignments will be graded on the edlab machines.

- If your code does not compile it will not be graded.
  - Comment out broken code.

- The grader will check your code for correctness as well as cheating and effort.
Working remotely

- We will be working in Unix on the edlab machines.
- If you are running Windows you will need to download PuTTY.exe.
- If you are running Linux or OSX you will need to use the ssh utility.
Using PuTTY

- Download and run PuTTY.exe
  - For host enter “elnux.cs.umass.edu”

- Enter your spire username and password.
  - If you don't have one see me after class.

- Congratulations you are now logged into an edlab unix machine!
Using ssh

- Open a terminal and enter:
  - ssh username@elnux.cs.umass.edu
  - Enter your password

- Congratulations you are now logged into an edlab unix machine.

- Note: if you can't connect try "username@elnuxD.cs.umass.edu"
  Where D is 1,2,3,4, or 7
Working in Unix

- `ls`
  - Lists all files in the directory.

- `pwd`
  - Names the current directory.

- `cd dir`
  - Moves you into the ‘dir’ directory.
  - `cd ..` Moves you ‘up’ a directory.
Working in Unix

- mkdir dir
  - Creates the 'dir' directory.

- rm filename
  - Deletes the named file.

- rm -r dir
  - Deletes the 'dir' directory.
Working in Unix

- `mv filename1 filename2`
  - Moves `filename1` to `filename2`.
  - Note: this deletes `filename1`.

- `cp filename1 filename2`
  - Copies `filename1` to `filename2`.

- `man cmd`
  - Displays help for using ‘cmd’.
Working in Unix

- make
  - Runs the make utility (which compiles C)

- ./exe arg1 arg2 arg3
  - Runs the ‘exe’ executable with arguments arg1, arg2, and arg3.

- echo $?
  - Displays the exit status of the last run executable.
To test your work type
  ./test

This will compile your code and run it.
Working in Unix

- To submit your work type
  - ./submit

- This will copy your work to your ‘submittedWork’ directory.
  - The work in this directory is automatically collected when due.
vim is a text editor.

To launch vim type “vim filename.c”
- This will open filename.c if it exists, or create it if it does not.

Here is a good tutorial for vim.
http://blog.interlinked.org/tutorials/vim_tutorial.html
vim: control mode

- When you open vim you can't type. You are in **control mode**.
- Control mode gives you access to the following commands
  - Xdd will cut X lines from the file.
    - These lines go to the clipboard.
  - Xp will paste the contents of the clipboard X times.
    - If X is excluded vim assumes X=1.
- You may use the delete key as usual.
vim: insert mode

- Press ‘i’ to enter **insert mode**.
  - In insert mode you can type directly in the document.

- When you are done typing press escape to return to **control mode**.
In **control mode** press ‘:’
- This moves your cursor to the bottom of the screen.

Here you can save and quit.
- `w` to save the file ('write')
- `q` to exit vim ('quit')
- `wq` to save and close
- `q!` to quit without saving changes
Finally writing in C

- Your first assignment is a basic calculator.
- You must edit a .c file that solves simple math problems.
- Detailed instructions are located INSIDE of Assignments/assignment1/calc.c
C functions

- To declare a function in C you write the following:
  - type name (arglist) {
    body
  }

- Just like Java!
C functions dissected

- type is the return type
  - This is a primitive

- name is the name of the function.
  - The function named ‘main’ is called when the executable is run.

- The arglist may be empty.
  - The arglist is formatted as in Java.
The main function

- int main(int argc, char * argv[]){
  body
}

- main returns an ‘exit status’.

- argc tells us how long argv is.

- argv contains the command line arguments.
Two notes about main

- **main** should return:
  - 0 (if it succeeds)
  - 1 (if it fails)
    - Technically we can return any int.

- `argv[0]` is the name of the executable.
  - Except when `argc` is 0, but this is rare.
Variables

- Some basic primitive types in C are:
  - int
  - char
  - float

- C treats char and int the same way:
  - So char aChar=5; is valid code.
  - [http://www.asciitable.com/](http://www.asciitable.com/)
Declaring variables

- To declare a variable write `type name;`
  - `type` is the type of the variable.
  - `name` is the name of the variable.

- Like Java, C will initialize your primitives for you.
  - This causes problems with **pointers** which we will talk about later.
Control Flow: if

- if( var < 5 ){
  Do something;
} else if( var < 10 ){
  Do something specific;
} else{
  Do something else;
}
Control Flow: for

- for( i = 0; i < 10; i++ ){
  do stuff;
}

- You **may NOT** declare a variable in your for statement. Ex:
  - for( int i=0; i<10; i++ ){
    do stuff;
  }

Control Flow: while

- `while( i < 10 ) {
  do something;
  i++;  
}

Control Flow: switch

- switch(varname){
    case A:
        do stuff
        break;
    ...
    default:
        do stuff
        break;
}

Your Assignment

- Write a .c file named calc that supports:
  - addition
  - subtraction
  - multiplication
  - division

- You can test your calculator using
  - ./test

- Be sure to submit your work using
  - ./submit