

# Introduction to Computer Science and Programming in C

Session 1: September 2, 2008

Columbia University

Administrative

# About the course

- General overview of Computer Science while learning how to program in C.
- Instructor: Bert Huang Tues. 2:30 - 4:30 PM CEPSR 624
  - TA: Deergha Sahni Wed. 3:00 - 5:00 PM TA room  
<http://ta.cs.columbia.edu/tamap.shtml>
- Tuesdays and Thursdays at 1:10 PM-2:25 PM
- <http://www.cs.columbia.edu/~bert/courses/1003>

# Requirements and Textbooks

- Basic computer skills
- CUNIX account
- **The C Programming Language** (2nd Edition) by Brian Kernighan and Dennis Ritchie
- **Practical C Programming** (3rd Edition) by Steve Oualline

# Course Policies

- Grading:
  - 4 homework assignments, 15% each.
  - In-class Midterm Exam, 15%
  - Final Exam, 20%
  - *Class Participation, 5%*

# Course Policies

- Grievances:
  - Type and print argument/ correction **on paper** and deliver to TA.
- Attendance and Reading
  - In your best interest.

# Academic Honesty

- Assignments in this course should be done individually.
- <http://www.cs.columbia.edu/education/honesty/>

# Courseworks

- We will be using the courseworks message board. <http://courseworks.columbia.edu>



# Who am I?

- PhD student in Computer Science department.
- I do Machine Learning research:
  - A mix of Artificial Intelligence and Statistics.
- I learned to program in C eight years ago.
- Still use it often when I need very efficient programs. For everyday research I use MATLAB.

# Who are you?

- Non-CS majors looking for programming skills.
- Potential CS majors (warning).
- You will learn C but you should be able to quickly pick up other programming languages.

Introduction to  
“Introduction to  
Computer Science and  
Programming in C”

# What is Computer Science?

- Scientific study of computers.
- “What can and can’t computers do?”
- “If a computer can do something, how can we do it efficiently?”

# What is a computer?

- A device that executes a sequence of computations and instructions.
- Modern computers are electronic and digital.
- Does pencil and paper count as a computer?

# Programs

- These sequences of instructions and computations is called a **program**.
- We'll be designing programs in this course.
- These programs will be based on **algorithms**.
- **Algorithm** - a step-by-step problem-solving procedure.

# Algorithms

- **Example Problem to solve:** Add a list of large numbers.

$$123 + 456 + 789 = ?$$

- Too much to compute in one step, so break down into smaller steps.
- $123 + 456 + 789 = 1,368$

# Algorithms

- Why are algorithms important?
  - Need to break down tasks into instructions computers are able to execute.
  - By defining algorithms, we can describe how to solve even more complex tasks.
    - $(123 + 456) + 789$



# Languages

- Computers operate on binary circuits:
  - **bits** are either on (1) or off (0).
  - Basic operations (adding, multiplying, etc).
- But we do not want to write our programs in 0's and 1's!
- Instead we use programming languages.

# Languages

- Want language to be close to English, but more precise.
- PCP describes legal system as attempting to program using imprecise English.
- Many disputes over interpretation of wording in legal text.

# Languages

- We will learn the C programming language.
- C is a “high-level” language; close to English.
- C was designed in the 1970’s by Dennis Ritchie for programming operating systems: Unix.
- C is very popular and is the basis for a few other popular languages:
  - C++, Java, Objective-C, C#

# Characteristics of C

- C is both praised and criticized for the amount of freedom given to the programmer.
- C is a **compiled** language
  - a compiler analyzes your code and translates it to efficient machine code.
  - We will be using the **open-source** GNU Compiler Collection or “gcc”

# Reading

- Read Chapter 1 in Practical C Programming.
  - Short introduction which basically goes over what we talked about today, in more detail in some parts and less detail in others.