CMSC 104

Problem Solving and
Computer Programming

Spring 2014

Instructor: Christopher Marron

First Things First…

First Things First…

Welcome!!!

First Things First…

Welcome!!!

First Things First…

Welcome!!!

First Things First…

Welcome!!!

(Especially new students!)

Instructor—Personal Information

- Who am I?
  - My name is Christopher Marron
  - New lecturer in CSEE, but taught STAT 355 at UMBC previous four semesters.
  - My day job: Visiting Professor in the CS Department at the Naval Academy.
  - I'm really a mathematician
    - 20 years at NSA: researcher, analyst, manager
    - Ph.D. in Mathematics

Instructor—Personal Information

- I’m interested in most areas of computation
- Previously worked in many areas of computer science, including high performance computing, machine learning, and language processing.
- Have spent most of my career in Government, but have taught for years.
- My goal for this class is to equip you to solve various analytic problems through programming.

Instructor—Personal Information

- Why do I teach?
  - I've always enjoyed teaching and have missed it when I've not been able to fit it in my schedule.
  - It appears that I'm reasonably good at it -- at least competent!
- Why CS?
  - I get bored. I taught stats for four semesters and wanted a change. I have have extensive experience in computing and enjoy programming.

Contact Information

- Best way to contact me?
  - Email! cmarron@umbc.edu
  - I will try for a 24-hour turnaround time, but please, no last-minute requests!
- Office hours:
  - Where?
    - TBD
  - When?
    - 7:00 - 7:45 pm, Monday and Wednesday

Apologies

- We (the CSEE Dept.) are confronting a resurgence of interest in the field
- Most of the undergraduate course offerings (and graduate, actually) are oversubscribed
- Long term: Good
- Short term: Bad

Am I in the Right Class?

- CMSC 104
  - Assumes NO programming experience
  - Introduces students to basic programming concepts like if/then structures and loops
  - Prepares you for CMSC 201
  - Does NOT count directly towards the CS major
  - Meets a requirement for other majors: i.e. Physics, Financial Economics

Am I in the Right Class?

- Advanced alternative: CMSC 201
  - Assumes some programming experience
  - First CMSC course for CS majors
  - Presumes you have basic grasp of thinking procedurally
    - E.g.: computer loops won't "throw you for a loop"
  - Focuses on more complex issues like proper design
  - Much more challenging

CS Minor Requirements

- Total of 23 credits (7 classes)
- Required courses:
  - CMSC 201 – Comp. Sci. I for Majors
  - CMSC 202 – Comp. Sci. II for Majors
  - CMSC 341 – Data Structures
  - CMSC 203 – Discrete Structures (can use MATH 301 instead)

CS Minor Requirements cont.

- Elective courses (9 credits):
  - 1 – 3 courses chosen from CMSC4xx (except 404, 496-498)
  - 0 – 2 courses chosen from:
    - CMSC 313 – Computer Org & Assembly
    - CMSC 331 – Principles of Programming Languages
    - MATH 221 – Linear Algebra

  (Note that this might change)

CS Game Development Track

- Web site: gaim.umbc.edu
- Not a separate degree just a "track" within the regular CS B.S. program
- Must complete all regular CS B.S. requirements plus some additional required and elective courses both in and outside the department (for example: "ART 380: History and Theory of Games")

What Will We Learn?

1. General computer hardware and software concepts
2. Basic computer use
3. Problem solving
4. Basic computer programming in the "C" programming language

1. General Hardware and Software Concepts

- Introduction to computer architecture
- Data representation and memory usage
- Introduction to operating systems

2. Basic Computer Use

- Basic use of
  - an operating system
    (Linux - new for most of you!)
  - a text editor (XEmacs)
  - a command-driven interface

3. Problem Solving

- Problem solving and algorithm development
  - general vs. specific solution to a problem
  - use of top-down design
  - use of pseudocode

4. Basic Computer Programming

- Creating and executing a computer program
- Testing and debugging a computer program
- C programming language basics

Course Information

- On the Web:
  - TBD!
- May use website *or* Blackboard *or* a combination of both. At the moment I don't have access to either!

Computer Science at UMBC

- CSEE Student Services (Advising)
  - ITE 203 - 206
- CSHC (Computer Science Help Ctr)
  - ITE 201-E
- Linux Users Group (LUG)
  - http://lug.umbc.edu
- Computer Science Council of Majors (CSCM)

Computer Labs

- The Division of Information Technology (DoIT) is responsible for all lab computers.
- On Web at:
  http://my.umbc.edu/topics/computing-and-technology
- Labs with PCs:
  - ENG021, ENG104, ENG122, ENG122A, ENG333
- Labs may be on reserve for classes, so plan ahead!
- Print Dispatch - ENG 019 (10? cents/page)
- Hours of Operations
  - DoIT will post outside of labs

DoIT Help Desk

- Located in ENG 020
- Phone: 410-455-3838
- Can help with a variety of things:
  - problems logging into your account
  - quota issues
  - communicating with UMBC computers (we'll discuss this in more detail later)
- Cannot help with course assignments

Computer Science Help Center

- CSHC is staffed by student tutors
  - Tutors available on a first-come-first-served basis
- Can help with
  - Homework and projects
  - XEmacs and Linux questions
- Located in ITE 201-E
  - Hours TBD

Hardware and Software Needs

- Do I need my own computer?
  - No, but it is more convenient for you.
- If I have my own computer do I need to install Linux?
  - No, you will be able to do your work in Windows (or on a Mac) as long as you have Internet access.

Using Your Own Computer: SSH

- We will discuss this in much more detail in future classes. You do not have to download anything at this point!!
- Windows users will need a software communications program like TeraTerm or PuTTY.
- Must be connected to the Internet.
- You can download TeraTerm from DoIT: http://my.umbc.edu/groups/doit/pages/2
- Consult DoIT for help.

Any Questions about Logistics?

That’s it for today.

Next time: Machine Architecture