Programming Methodology

Lecture 01 - Introduction

UMASSCS

SCHOOL OF COMPUTER SCIENCE

Today's Objectives

- **Introductions**
- **Communication**
- **Course Objectives**
- **Course Schedule**
- **Software Environment**
- **Assignments and Grading**
- **Basic Command Line**
- **Introduction to Version Control**

Introductions - Instructor

Instructor

Tim Richards
richards AT cs DOT umass DOT edu
Computer Science Building Room 258
Office Hours: We 10:00AM – 11:00AM
By Appointment

Introductions - Teaching Assistants

Teaching Assistants

Brendan Murphy
bemurphy AT cs DOT umass DOT edu
Office Hours and Location: TBD

Theodore Sudol
tsudol AT cs DOT umass DOT edu
Office Hours and Location: TBD

Introductions - Graders

There are a number of graders for the course. They will remain anonymous, but they will be a crucial component of the 220 staff. They will be responsible for reviewing and grading your work. They are top students who have taken the course previously. If you know who they are please do not ask them about 220!

Course Resources

Official Course Website
http://cs220.weebly.com
Piazza
https://piazza.com/umass/spring2015/cmpsci220/home
Moodle
http://moodle.umass.edu
Gitblit
https://elnux1.cs.umass.edu:8443

Communication - Class Meetings

You are expected to come to class.

If you miss class you are responsible for finding out what you missed. We use class as the most important place for communication and assume you are in attendance. If you missed important details communicated during class that are not posted to Piazza it is not our responsibility.

Key Point: Come to every class unless you are truly ill or for a good enough reason justifying your absence. Make sure you know exactly what was covered and what was communicated.

Communication - Course Website

You will find all the information you need here.

- Course Description, Syllabus, Schedule, etc.
- Links to piazza and moodle
- Links to assignments, class material, etc.
- Office hours and location

We will assume you read this in detail!

Communication - Piazza

First Stop for Questions/Answers

Piazza is a wonderful tool when used correctly. It will allow you to ask questions about topics that are covered, get answers to those questions, answer questions from other students, and continue the dialogue from class. It will be monitored closely by the course staff.

Piazza is not a replacement for class. You should not expect answers to questions that were clearly stated in class. You should expect a reasonable amount of time for a response. You should not necessarily expect a response from course staff. You should not ask questions before **thinking** first.

Communication - Moodle

A Place to Find Assignment and Grade Info

Assignments will be posted from the course website, however, Moodle will maintain the official due dates and you will be expected to submit your work to Moodle. You will take online quizzes in Moodle. You will also be performing peer review of other student's work in Moodle. You will find all your grade information for the course in the Moodle gradebook. We will not be using Moodle for online discussions - for that, go to Piazza.

Communication - Gitblit

A Place to Practice Version Control

We will be using the Git version control system in this course. You will be required to submit your work through Git and you will be graded on your effective use of Git and your activity level.

Gitblit is an online hosting system that we are running in this course. You will be provided a remote repository where you will place all of your assignment work. You will be required to submit your work using both Git and Moodle. Assignments will clarify what you must do.

Gitblit is my experiment this semester - if you find that the site is down you should contact the course staff (Piazza) immediately.

Communication - Office Hours

A Place to Ask Questions in Person

Office hours will be held by the Instructor and Teaching Assistants at various times during the week. You are welcome to visit office hours and ask questions about the course material, assignments, etc. Here are some tips for office hour visits:

- Come prepared - have an idea of what you want to talk about.
- Do not expect to code your assignment in front of us.
- Be respectful of our time and understand that office hours have an end.
- Try before you come. Do not show up empty handed.
- Do not expect to be taught material after missing class.
- Communicate effectively so we understand your problems.
- Drive the conversation, we will not guess why you are there.

Communication - EMail

An Extreme Last Resort

EMail is a very useful tool. At the same time it has a tendency to be **overused**. Given the large amount of email received over the course of a day it is often the case that your message may go unnoticed for days or even weeks if you send it to just one of us. You should post to Piazza to ensure that everyone on the course staff gets your message. You can post to Piazza privately to instructors and can be ensured that we will receive your message. Again, do not abuse this either.

Communication - Response Time

We Try to Do Our Best, but...

It is easier than you think that we miss your posts or email. This is not personal - we are only human. Please give us a reasonable amount of time to respond to you. For example, do not post to Piazza at 3AM and expect us to respond immediately (or even that morning or day). Do not send repeated posts unless a sensible amount of time has passed and you have not received a response. Just like EMail, Piazza can easily become hard to manage with several repeated posts.

Make sure you read Piazza to see if your questions has already been answered. If you ask a duplicate question you may not get an answer.

i-clicker question

Have you used the Java programming language?

a) Yes
b) No

i-clicker question

How well do you know Java?

a) I am a ninja master
b) I am pretty good
c) I am ok, but not great
d) I am very weak
e) I could not code myself out of a paper bag

i-clicker question

Have you used a programming language other than Java?

a) Yes
b) No

i-clicker question

Have you used either Python/Ruby/Perl/JavaScript?

a) Yes
b) No

i-clicker question

Have you used the C programming language?

a) Yes
b) No

i-clicker question

Have you used a *functional* programming language?

a) Yes
b) No
c) I don't know

i-clicker question

Do you know the difference between *functional*, *object-oriented*, and *imperative* programming languages?

a) Yes
b) No
c) A little bit

i-clicker question

Have you used a text editor to program (not an IDE)?

a) Yes
b) No
c) A little bit

i-clicker question

Have you used *version control* before?

a) Yes
b) No
c) A little bit

Course Description

Development of individual skills necessary for designing, implementing, testing, and modifying larger programs, including: use of integrated development environments, design strategies and patterns, testing, working with large code bases and libraries, code refactoring, and use of debuggers and tools for version control. There will be significant programming and a midterm and final examination. Prerequisites: CMPSCI 187 or ECE 242.

What does this mean?

Course Objectives - Overview

From Novice to Advanced

This course aims to bring you from a **novice programmer to an advanced level** (note - not expert). The world of programming and software development is filled with many languages, tools, and techniques that **require study and experience** to become an *effective programmer*. Our goal in this course is to expose you to many of these topics and raise your level of understanding in programming and software development to prepare you for future courses (e.g., 320, 326). In addition, we want to ignite a flame in your programmer mind and soul that will enable you to learn on your own and seek out knowledge in programming and languages for **your own self edification and advancement**.

Course Objectives - Chapters

A Story to Follow...

Details can be found on the course schedule.

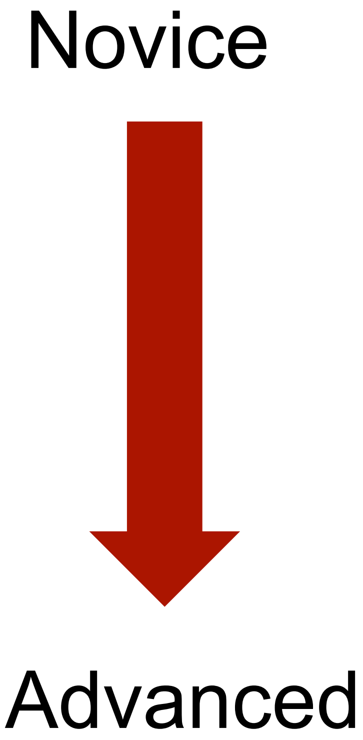
- Chapter 1: Programming Principles
- Chapter 2: Programming Paradigms and Patterns
- Chapter 3: Programming in the Large

Specific Course Objectives…

Can be found in the course syllabus

Weekly Schedule

Week 01: Introduction
Week 02: Principles of Good Programming
Week 03: Programming Paradigms
Week 04: Functional Programming
Week 05: Immutability
Week 06: Composition and Inheritance
Week 07: Design Patterns
Week 08: Regular Expressions
Week 09: Libraries
Week 10: Parallelism Patterns
Week 11: Data Patterns
Week 12: Domain Specific Languages
Week 13: Conclusion



Novice

Advanced

Software Environment

We will be using VirtualBox/Vagrant
We will be using the JVM (Java/Scala).
We will be using Git.
We will be using the command line and basic text editors.
We will be using IDEs (Eclipse/IntelliJ)

You are welcome to use any environment you wish (Windows, Mac, Linux). We will provide a VirtualBox/Vagrant environment that will allow you to run Linux with most of the software installed. Part of this course is about figuring out how to configure your software environment. We will not offer direct instruction on how to do this - this is part of the challenge. We will provide links to the software to install.

Assignments and Assessment

- **Project Assignments**
  - There will be several
- **Project Exams**
  - There will be 2-3
- **Peer Reviews**
  - Accompany most project assignments
- **Other Activities**
- **Participation**

Assignment Submission

You will be required to submit your project assignments through both Git/Gitblit and Moodle. We will use Gitblit to evaluate your understanding of version control and your activity level. We will use Moodle for peer review and grade association. You must submit to both for your submission to be complete. A missing component will negatively impact your grade on the assignment.

Assignment Grading

You will be graded on a number of aspects including functional (did it pass tests), your own tests, code style, following instructions, git activity, comments and documentation, and any other criteria designated by the assignment documentation.

Grades will be on a letter scale: A-F

This course does not give 0's on assignments. **Why?**

Because: 100+0/2 = 50 (an F), 100+50/2 = 75 (a C)

But: A+F/2 != F so it makes sense A+F/2 = C (in my book!)