Computer Systems Principles

Course Administration & Introduction

UMASSCS

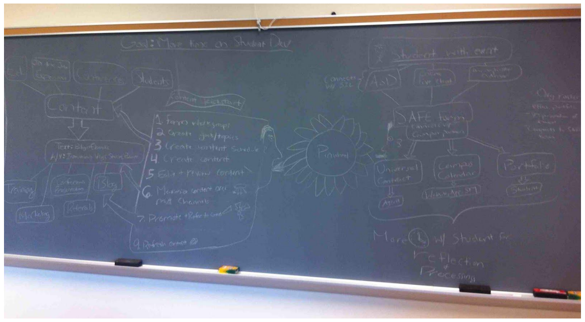
SCHOOL OF COMPUTER SCIENCE

Today

- **Course Administration**
  - Structure
  - Grading
  - Academic Honesty and MOSS
  - Programming Environment
  - How to be successful in this course
- **Introduction**
  - What is this course about?
  - Topics we will cover

Course Structure

- **Lectures**
  - Highlight important concepts and techniques
  - Case studies with code & in-class activities
  - i>clicker (one measure of participation)
- **Discussion ---** exercises to solidify your understanding

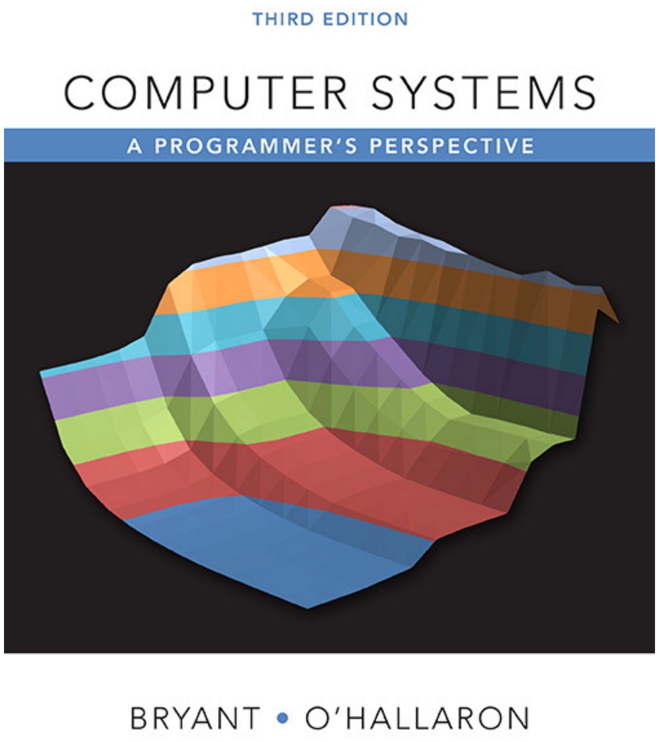


Textbook

- **Computer Systems: A Programmer's Perspective, 3E** by Bryant and O'Hallaron

- We will not cover this book in its entirety, only the material we focus on. This includes parts of chapters 1, 2, 3, 7, 8, 9, 11, & 12

- Available through the normal channels



Course Staff

Instructors

Kaituo Li

John Ridgway

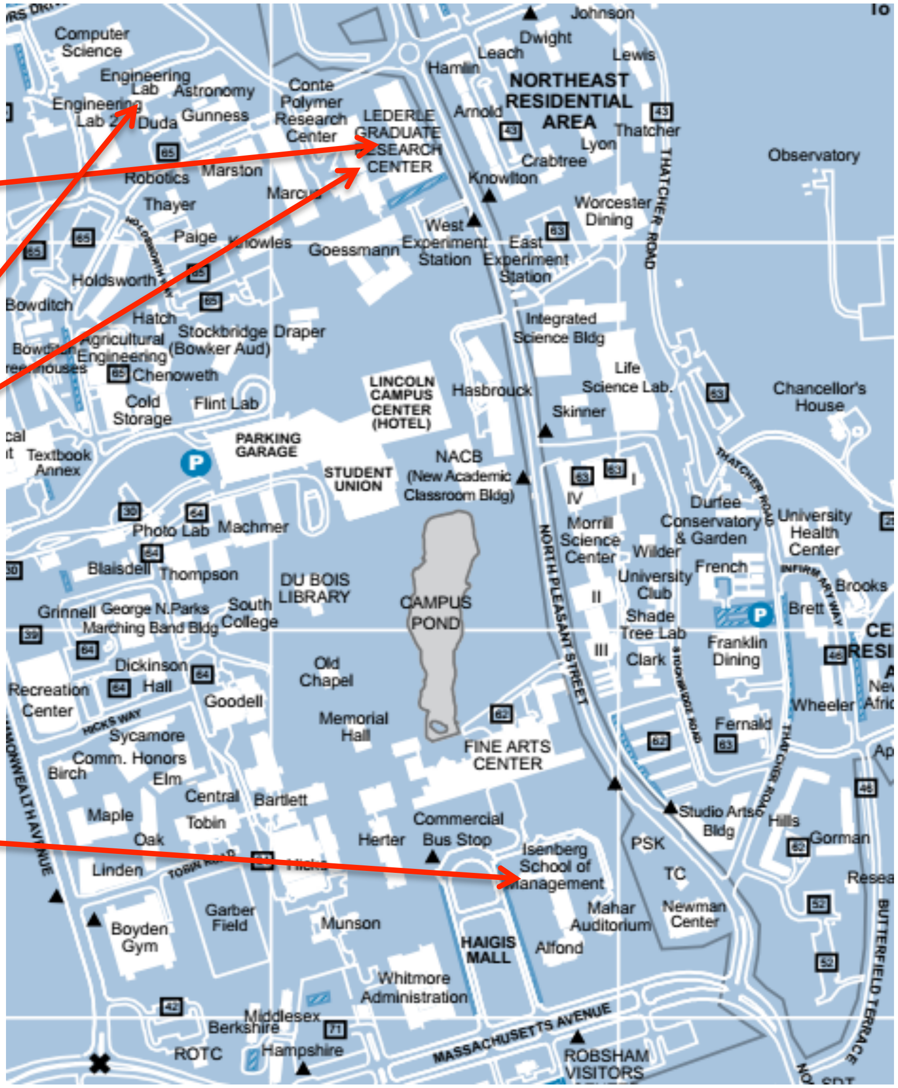
Teaching Assistants

Nick Braga

Brendan Teich

Orientation

- **Office Hours**
- **Discussions**
- **Lectures**



Course Resources

- **Website**
  - Course material, assignments, syllabus, schedule, etc.
  - http://www-edlab.cs.umass.edu/cs230
- **Moodle**
  - Communication with course staff and other students
  - Assignment submissions, quizzes, exams
  - Please check regularly
  - If you are registered for the course you should see it in Moodle https://moodle.umass.edu

Communication

- Moodle forums, one for announcements, one for discussion
- Post your questions and answer other people's questions
- TA and instructors will be monitoring and will try to respond quickly
- **DO NOT POST CODE.**

Assessment/Grading

Weight	Assessment
50%	Project Assignments
10%	Quizzes
5%	Participation
15%	Midterm Examination
20%	Final Examination

A plan: subject to adjustment

Project Assignments

- This course has several project assignments
- Each project is a programming assignment designed around the topics we are studying
- Almost all project assignments must be implemented in the C Programming Language
- All project assignments must be completed individually
- Project assignments will be **auto-graded** as much as possible

Quizzes

- Quizzes will be taken <u>online</u> through Moodle.
- Each quiz will be based on the material covered that week. Some question may require research on the web.
- You will have a week to complete the quiz. Quizzes are **not timed**. You may revisit a quiz at any point during the week. These are low stakes: they count 10% *in total*.
- **Adaptive**: questions answered incorrectly on a quiz can be attempted again with a penalty applied (except T/F).
- Quizzes will be automatically submitted when the due date expires. We will not allow retakes/reopening of quizzes after that time.

Discussions

- Discussion sections must be attended
- Discussion will primarily consist of exercises for active learning.
- i>clickers will be used to collect attendance and to check understanding

Participation

- i>clicker exercises (responding matters; the specific answers do not!)
- Forum contributions

Lateness Policy

- Late assignments will not be accepted without a documented excuse.
  - That's for your benefit and for ours.
- Absolutely no credit given after solutions posted
- Extensions may be granted in extenuating circumstances if requested sufficiently in advance (usually 24 hours)

Academic Dishonesty

- We take this very seriously. It will have a negative impact on your course grade, your GPA, and perhaps your overall record at Umass.
- You may discuss assignment problems with others in the course; however, writing (including code) of solutions must be your own.
- Copying any material directly or indirectly from the web is considered dishonest – even if it is GPL, Apache, BSD, MIT, or any type of open source license.
- Copying or using sections of someone else's program or assignment, even if it has been modified by you, is dishonest.
- We will be using the MOSS system to detect software plagiarism.
- When in doubt, ask whether it's OK.
- Please see the UMass Code of Conduct for further details
- Multiple students have received an F in this course for dishonesty. Follow the rules so it won't happen to you!

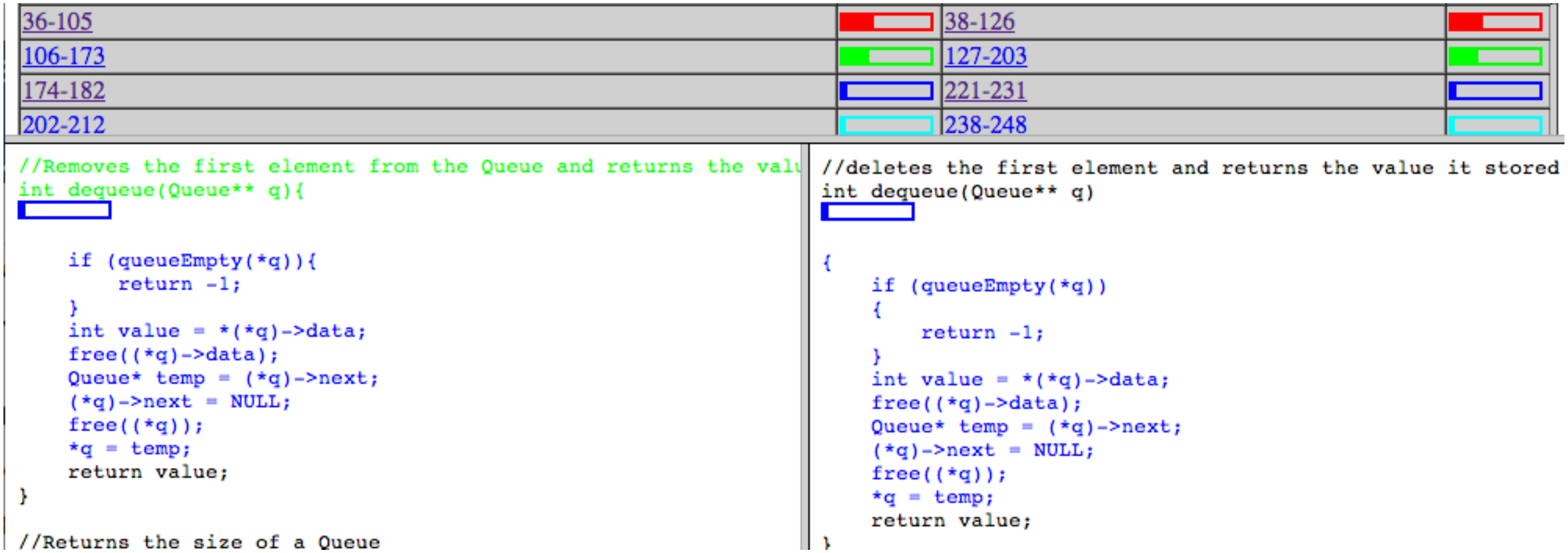
Examples of Academic Dishonesty

- Viewing all or part of your friends assignment source code either in person or over Skype, Google Hangout, Facetime, or any other video or audio transmission.
- Sending all or part of your assignment source code through email, text, or Morse code.
- Posting all or part of your assignment source code on the web – this also means online public source code repositories such as github or bitbucket.
- Posting all or part of your assignment source code on the Moodle forums.

More Examples of Dishonesty

- Googling for solutions to an assignment.
- Copy and paste solutions from the web into your editor, even with heavy editing, and submitting it as your own.
- Using solutions from previous semesters.
- Paying someone to do your assignment and submitting as your own.
- Stealing a friend's source code with a thumb drive (yes, this has happened).
- Stealing from your roommate's computer during spring break (actually happened).

MOSS Plagiarism Detection



36-105	🟥	38-126	🟥
106-173	🟩	127-203	🟩
174-182	🟦	221-231	🟦
202-212	🟦	238-248	🟦

```
//Removes the first element from the Queue and returns the valu
int dequeue(Queue** q){


    if (queueEmpty(*q)){
        return -1;
    }
    int value = *(*q)->data;
    free((*q)->data);
    Queue* temp = (*q)->next;
    (*q)->next = NULL;
    free((*q));
    *q = temp;
    return value;
}

//Returns the size of a Queue
```

```
//deletes the first element and returns the value it stored
int dequeue(Queue** q)


{
    if (queueEmpty(*q))
    {
        return -1;
    }
    int value = *(*q)->data;
    free((*q)->data);
    Queue* temp = (*q)->next;
    (*q)->next = NULL;
    free((*q));
    *q = temp;
    return value;
}
```

Do not even bother! The above submission from a recent semester was 82% similar. In a typical case only 25% is an almost certain indication that plagiarism took place. If you don't believe me read the posted paper.

Think before you do something stupid

- If you are suspected of academic dishonesty you have an opportunity to refute the evidence from MOSS. If you know you are lying, don’t bother, you will only embarrass yourself and dig a deeper hole.
- If we determine that academic dishonesty did take place you will **receive an F** for the course and an informal resolution will be filed with the UMass Academic Honesty Office – or you can pursue a formal hearing.
- If you refuse to meet with us then we will file a formal charge and a hearing will be scheduled where evidence will be presented. You will lose. Consequences may be worse than an F for the course.
- About 20 cases the semester before last received an F.

Think before you do something stupid

i>clicker question

Which is the following is *not* dishonest?

A – Googling the answer

B – Using code from a previous semester

C – Taking a friend's code and modifying it

D – Emailing your code to a struggling friend

E – None of the above

Programming Environment

- VirtualBox
- Lubuntu
- C Programming Language
- gcc, gdb, objdump, make, …
- Editors

Why people fail this course?

- **Phrasebook C**
  - Need to know how to use C, not just what it looks like
- **Not knowing what you don't know**
  - Come to class and discussion sections
- **Learning material by osmosis**
  - This course is not simply a shot of knowledge in the head, and we don't spoon feed everything
- **Not practicing enough**
  - Knowing the syntax is important, but not enough
  - Practice, practice, practice!

What is this course about?

Computer Systems Principles

Is about the different layers in a computer system…

Computer Systems Principles

Is about the different layers in a computer system…

Computer Systems Principles

C Programs & Libraries
Assembly
Machine Code
Virtual Memory
Memory (Real Memory)
Information Representation
Processor

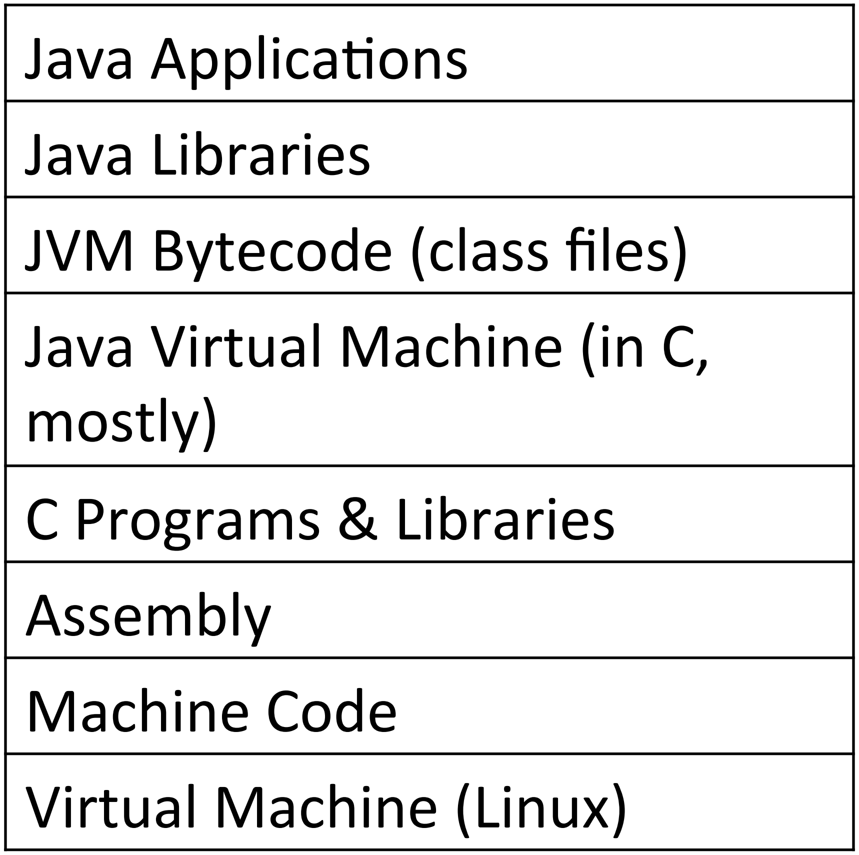
Is about the different layers in a computer system…

Group Activity

- **Where does Java fit in this mess?**
  - Take 5-10 minutes to form a group and discuss these questions with the people in that group:

1. How does a Java program execute?
2. Does the machine execute Java source code?
3. What does Java rely on to run on a modern OS?
4. How is Java implemented and what does it need in order to even exist?
5. What is C and how does it compare to Java?
6. What is a Java Virtual Machine (JVM)?

Computer Systems Principles



Java Applications

Java Libraries

JVM Bytecode (class files)

Java Virtual Machine (in C, mostly)

C Programs & Libraries

Assembly

Machine Code

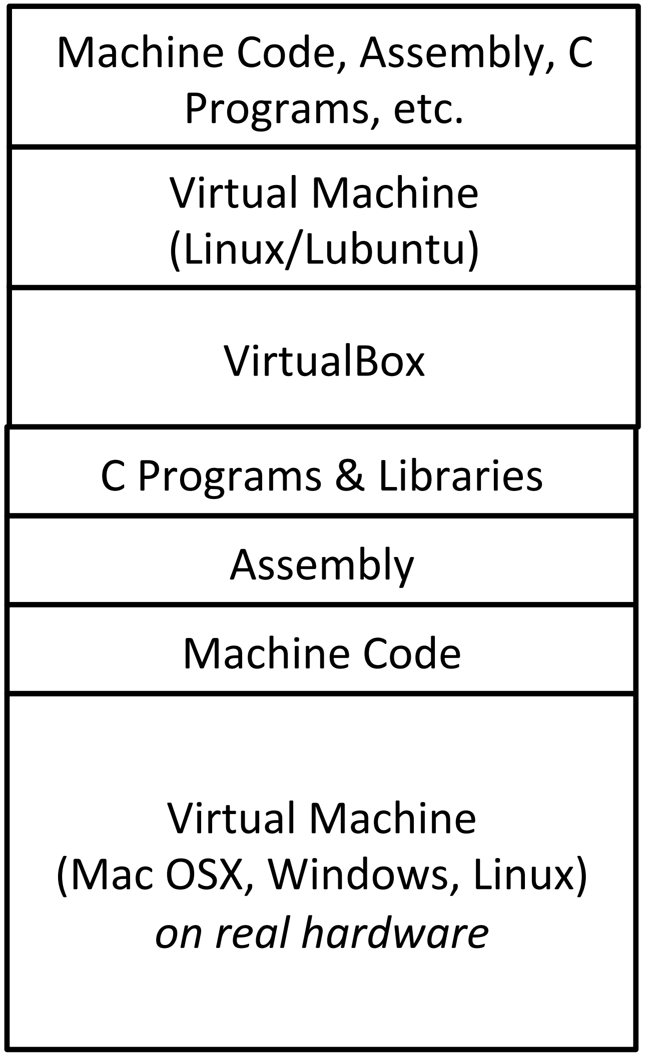
Virtual Machine (Linux)

Is about the different layers in a computer system…

Programming Environment

- **VirtualBox**
  - A user application that emulates an x86_64 microprocessor and hard disk
  - Once an OS is installed on the emulated hard drive, it acts like a whole computer
  - We will use this to run a virtual Linux machine
  - Where does this go in the stack?

VirtualBox Stack



Machine Code, Assembly, C Programs, etc.

Virtual Machine (Linux/Lubuntu)

VirtualBox

C Programs & Libraries

Assembly

Machine Code

Virtual Machine (Mac OSX, Windows, Linux) *on real hardware*